AD-A035 911 NAVAL POSTGRADUATE SCHOOL MONTEREY CALIF F/G 20/11
STATIC AND DYNAMIC BUCKLING OF SHALLOW SPHERICAL SHELLS SUBJECT--ETC(U)
DEC 76 M D SHUTT
UNCLASSIFIED NL

1 OF 2
AD-A
035 911

U.S. DEPARTMENT OF COMMERCE
National Technical Information Service

AD-A035 911

STATIC AND DYNAMIC BUCKLING OF SHALLOW SPHERICAL SHELLS SUBJECTED TO AXISYMMETRIC AND NEARLY AXISYMMETRIC STEP-PRESSURE LOADS USING SATANS-IIA, A MODIFIED VERSION OF SATANS-II

NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIFORNIA

DECEMBER 1976

NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

STATIC AND DYNAMIC BUCKLING OF SHALLOW SPHERICAL SHELLS SUBJECTED TO AXISYMMETRIC AND NEARLY AXISYMMETRIC STEP-PRESSURE LOADS USING SATANS-IIA, A MODIFIED VERSION OF SATANS-II

by

Michael D. Shutt

December 1976

Thesis Advisor: Robert E. Ball

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Static and Dynamic Buckling of Shallow Spherical Shells Subjected To Axisymmetric and Nearly Axisymmetric Step-Pressure Loads Using SATANS-IIA A Modified Version of SATANS-II		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis; December 1976
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Michael D. Shutt		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		12. REPORT DATE December 1976
		13. NUMBER OF PAGES 167
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Buckling
Shallow Spherical Shells
Axisymmetric
Asymmetric
Computer Program

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A digital computer program for the geometrically nonlinear analysis of totally arbitrarily loaded shells of revolution (SATANS-II) was modified to more accurately account for the conditions at the pole of the shell. This program was used to determine the buckling load of shallow spherical shells of various sizes when subjected to static axisymmetric, dynamic axisymmetric, and nearly axisymmetric step-pressure loads of infinite duration. A comparison was made between the new buckling results and previous results obtained without the new pole routine. The comparison revealed a significant

20. (continued)
change in the buckling pressures, due solely to the change in the pole routine. The new static axisymmetric, dynamic axisymmetric, and even the dynamic asymmetric critical buckling pressure loads appear to be fairly reliable results for perfect, shallow shells.

STATIC AND DYNAMIC BUCKLING OF SHALLOW SPHERICAL SHELLS SUBJECTED TO AXISYMMETRIC AND NEARLY AXISYMMETRIC STEP PRESSURE LOADS USING SATANS-IIA, A MODIFIED VERSION OF SATANS-II

by

Michael D. Shutt
Lieutenant
B.S., Oregon State University, 1970

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN AERONAUTICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL

December 1976

Author: Michael D. Shutt

Approved by: Robert E. Ball, Thesis Advisor

Milton H. Bank, Second Reader

Richard W. Bell, Chairman, Department of Aeronautics

Robert R. Fossum, Dean of Science and Engineering

ABSTRACT

A digital computer program for the geometrically nonlinear analysis of totally arbitrarily loaded shells of revolution (SATANS-II) was modified to more accurately account for the conditions at the pole of the shell. This program was used to determine the buckling load of shallow spherical shells of various sizes when subjected to static axisymmetric, dynamic axisymmetric, and dynamic nearly axisymmetric step-pressure loads of infinite duration. A comparison was made between the new buckling results and previous results obtained without the new pole routine. The comparison revealed a significant change in the buckling pressures, due solely to the change in the pole routine. The new static axisymmetric, dynamic axisymmetric, and even the dynamic asymmetric critical buckling pressure loads appear to be fairly reliable results for perfect, shallow shells.

TABLE OF CONTENTS

LIST OF SYMBOLS

b = nondimensional inplane stiffness

E = the modulus of elasticity of the shell

H = the rise of the spherical cap at the pole

h = the thickness of the shell

m = the mass density of the shell

M_s = the meridional bending moment per unit length

n = the Fourier index

P = a nondimensional applied load

P_{CRIT} = the nondimensional critical pressure

q_o = the classical buckling pressure of a complete sphere

$q^{(n)}$ = a column matrix containing the coefficients of the n^{th} term in the series expansion of the applied load

r = the normal distance from the axis of revolution to the surface of the cap

r_o = the normal distance from the axis to the cap in the base plane; the maximum value of r

R_s, R_θ = the radii of curvature in the s and θ directions, respectively

s = the meridional distance along the surface of the shell

t = the nondimensional time

T = the time

T_o = a reference time

U, V, W	=	the displacements in the s, θ and ζ directions, respectively
u, v, w	=	nondimensional series coefficients of U, V, W
$\bar{V}$	=	a nondimensional measure of the volume of the shell deformation
$\bar{V}_{MAX}$	=	the peak in the time history of the parameter $\bar{V}$
$W^{(n)}$	=	the displacement in the ζ direction in the n^{th} harmonic
δt	=	the nondimensional time increment
	=	distance between stations
$\epsilon^{(n)}$	=	the nondimensional parameter governing the magnitude of the load applied in the asymmetric harmonics
ζ	=	the coordinate normal to the surface of the shell
θ	=	the circumferential angle measured about the axis of revolution
λ	=	a nondimensional geometric parameter used to describe the spherical cap
ν	=	Poisson's ratio
ξ	=	the normal distance from the base plane to the middle surface of the undeformed cap

ACKNOWLEDGEMENT

I wish to acknowledge the help that I have received in the course of my thesis research.

I am particularly indebted to Professor Robert E. Ball, who provided me with the guidance and aid necessary to complete my project.

I am also grateful to Professor Johann Arbocz of CALTECH, who provided the initial deck of cards from which I worked.

I also wish to give credit to the NPS programming consultants, and in particular Richard Donat, for the expert assistance he provided in debugging my changes to the computer program.

I especially want to express my sincere appreciation to my wife, Roberta, for her patience and assistance in typing of this thesis.

I. INTRODUCTION

In 1973 a digital computer study was presented by Ball and Burt [1] for the dynamic buckling load of clamped shallow spherical shells subjected to axisymmetric and nearly axisymmetric step-pressure loads. A static buckling analysis of the same spherical shells had been carried out in 1970 by Stilwell and Ball [2]. In these two studies the digital computer program SATANS-I [3] was used to calculate the critical buckling pressures for a large range of shell sizes. Other studies of the buckling of shallow shells have been conducted by Huang [4,5], by Stephens and Fulton [6], by Lock et al. [7], by Stricklin [8], and most recently by Akkas [9]. In Reference 1 the results from these other studies, except for those by Akkas, are compared with the results from SATANS-I for both static and dynamic buckling. In the axisymmetric static analysis the comparison with the results obtained by Huang [4] revealed that the SATANS-I results were higher than Huang's results for several shell sizes. In the dynamic, axisymmetric buckling analysis the SATANS-I results again either agreed closely with, or were somewhat higher than , the results by Huang [5], Stephens and Fulton [6], and Stricklin [8]. However, it was noted then that there was a general lack of consistent agreement among any of the sets of results. As a consequence, it appeared at that time that the axisymmetric buckling problem had not yet been totally resolved and that additional studies would be appropriate.

In the asymmetric dynamic buckling analysis of Reference 1 the few comparisons that could be made for the critical load also indicated that the SATANS-I results may be too

high. A comparison of the recent estimates for the asymmetric dynamic buckling load obtained by Akkas [9] with the SATANS-I results also reveals the SATANS-I results to be well above those of Akkas [9]. However, it should be noted that the results obtained by Akkas were from his attempt to obtain a lower bound on the critical asymmetric load. This bound on the buckling load is obtained without the execution of a complete transient response analysis on the asymmetric part of the response of the shell, as is done in SATANS-I. In Akkas' analysis (Problem 1) the transient nonlinear axisymmetric response is computed, and a determinant is examined for possible bifurcation into asymmetric motion at each time step. The minimum load at which the determinant becomes zero is defined as the lower bound of the critical load.

As a consequence of the generally high buckling loads predicted by SATANS-I, a re-examination of the static and dynamic buckling of the shallow spherical shell was made in an attempt to determine the possible cause, or causes, of the high buckling loads. In our search we discovered that a modification of the manner in which the pole conditions are numerically approximated significantly lowered the buckling loads to values that are now in good agreement with the other results. The new procedure for handling the pole condition is given in section III of this thesis. The new buckling results are given in section V.

In addition to the pole condition modifications and the new buckling results the author has also made another significant change to the SATANS family of codes. In particular, the SATANS-II program for the geometrically nonlinear analysis of totally arbitrarily loaded shells of revolution, developed by Ryan [10] in 1972 to handle more complex and larger problems, was modified to make the computer memory requirement a variable quantity. This

quantity is specified by the user to fit the particular problem being run. It eliminates the large core requirement of SATANS-II for small problems and allows for much larger problems to be solved than could be solved by SATANS-II. The new program with the pole condition and memory modifications will hereafter be called SATANS-IIA. It is described in section II.

II. DESCRIPTION OF SATANS-IIA

SATANS-II was developed by Ryan [10] from SATANS-I and incorporated the full trigonometric expansion of the applied load and solution vector, and introduced the handling of imperfections into the code. These modifications allow the analysis of shells under totally arbitrary loads, as well as imperfection studies on actual shells with measured imperfections [11]. Unfortunately, the original deck of cards for SATANS-II was destroyed. Professor Johann Arbocz of CALTECH had a listing of SATANS-II and punched a deck of cards with the changes to SATANS-I given in that listing. A copy of this deck was sent to Professor Ball. These cards have been added by the author to the original SATANS-I described by Ryan [10] and a complete version of SATANS-II has been reconstructed. SATANS-IIA is a modification by the author of the reconstructed SATANS-II program. A listing of SATANS-IIA can be found in Appendix A. The listing contains an example problem for the dynamic analysis of a clamped, truncated cone subjected to an impulsive loading which is uniform along the meridian and varies in a cosine distribution over one-half of the circumference. This problem is a sample problem suggested by the Lockeed Missiles and Space Corp. [12]. A condensed version of the output from the example problem is given in Appendix B. Input data preparation for SATANS-IIA can be found in Appendix C. The basic users manual, which includes preparation of input subroutines and the theory of the program, is contained in Reference 3, which can be obtained through COSMIC (M70-10098, LAR-10736), or ASIAC [13]. A users manual which includes preparation and handling of imperfection data within the SATANS programs can be found in

Ref. [10]. The above information, along with the following discussion, will inform the user on the capabilities and proper use of SATANS-IIA.

The modification of the SATANS-II program to make its core requirement variable was accomplished by putting in a single dimension statement at the beginning of the program, with subsequent dimensioning within the subroutines to only the first element of the vector or matrix. This is a convenient feature of the FORTRAN-IV language in which the program is written. The actual vector and matrix sizes are transmitted to the subroutines by an individual parameter list. Construction of the initial dimension statement and core request size is as follows:

The basic size of the program on the IBM-360/67 Digital Computer, without the initial dimension statement, is 272,000 bytes. This figure includes approximately 19,000 bytes of buffer space required for execution. Within the main dimension statement are fifteen variables. However, only three parameters are needed to specify the sizes of these fifteen variables.

Let a= The number of stations along the meridian of the shell times the number of harmonics considered.

Let b= a, plus two fictitious stations times the number of harmonics considered.

Let c= The number of harmonics considered.

The main dimension statement would then be constructed as,

```
DIMENSION P(4,4,a), DEE(4,4,a), DST(4,4,a), X(4,a),
          PHIXB(a), PHITB(a), Z(4,b), ZO(4,b),
          Z2(4,b), Z3(4,b), ZDOT(4,b), IS(99,c),
          JS(99,c), ID(99,c), JD(99,c)
```

The 99's above limit the user to 99 harmonics in any one run and an unlimited number of meridional stations. The core requirement for the general case would be,

272,000 + 216a + 80b + 1584c = bytes of core required.

For a sample calculation of the core requirements consider the example of a spherical cap with 40 stations along the meridian, and an asymmetric analysis with two harmonics. Therefore,

a= 40(stations) x 2(harmonics) = 80

b= 80 + 2 x 2(harmonics) = 84

c= 2(harmonics)

Thus, for the variables P, DEE, DST,

3 x (4x4x80) = 3840 (words) x 4 = 15,360 bytes

for the variable X,

4 x (80) = 320 (words) x 4 = 1280 bytes

for the varibles PHIXB, PHITB,

2 x (80) = 160 (words) x 4 = 640 bytes

for the varibles Z, ZO, Z2, Z3, ZDOT,

5 x (4x84) = 1680 (words) x 4 = 6720 bytes

lastly, for the varibles ID, JD, IS, JS,

4 x (99x2) = 792 (words) x 4 = 3168 bytes

Therefore, the total size of the main dimension statement would be 27,168 bytes. This figure would be rounded up to the nearest even thousand bytes, i.e. 28,000 bytes. Finally, the core requirement for this example problem would be

272,000 + 28,000 = 300,000 bytes.

III. IMPROVED POLE ROUTINE

The SATANS code is based upon Sander's geometrically nonlinear equations under the conditions of small strains and moderately small rotations. The formulation is in four second order nonlinear partial differential equations in terms of U, V, W, and M_s, where U, V, and W are the meridional, circumferential and normal displacements respectively, and M_s is the meridional bending moment. The nonlinear partial differential equations in the coordinates s, θ, and t are reduced to uncoupled sets of linear differential equations in s and t by expanding the variables in trigonometric series in the circumferential coordinate θ, and treating the nonlinear terms as pseudo loads. The first and second derivatives in the meridional coordinate s are replaced by the conventional central finite difference approximations, i.e.

$$\{z\}'_i = 1/2\Delta \left(\{z\}_{i+1} - \{z\}_{i-1}\right) \tag{1}$$

and

$$\{z\}''_i = 1/\Delta^2 \left(\{z\}_{i+1} - 2\{z\}_i + \{z\}_{i-1}\right) \tag{2}$$

where $\{z\}_i$ is the vector of U, V, W, and M_s at the i^{th} station, Δ is the uniform dimension between stations, and primes denote partial derivatives with respect to s. Applying these approximations to the governing set of domain

equations leads to

$$[C]_i \{z\}_{i-1} + [B]_i \{z\}_i + [A]_i \{z\}_{i+1} = \{g\}_i \tag{3}$$

When the shell does not have a pole, fictitious stations one increment off of the shell are introduced at each end. Both the governing domain equations and the boundary conditions are applied at the two boundary points. Thus, all finite difference approximations to the derivatives, including those of the boundary conditions, are of order Δ^2. However, prior to the development of SATANS-IIA, the treatment of the conditions to be applied at a pole at either end of a shell was handled by a simple Euler forward or backward difference approximation to the first derivative, with truncation error of order Δ. For example, for a pole at s= 0, where i= 1, the first derivative at the pole was approximated with

$$\{z\}'_1 = 1/\Delta \; (\{z\}_2 - \{z\}_1). \tag{4}$$

At the time this procedure for handling the pole conditions was developed (1967) it was thought that this would not significantly alter the solution. However, it has since been discovered that such is not the case.

For the new pole routine, an expanded forward difference approximation of order Δ^2 is used at s= 0 which takes into account the two stations after the pole, instead of just one station after the pole as in the Euler scheme. This approximation is

$$\{z\}'_1 = 1/2\Delta \; (-3\{z\}_1 + 4\{z\}_2 - \{z\}_3). \tag{5}$$

The conditions to be imposed upon the dependent variables at a pole are derived in Reference 14. They are :

For N= 0, $u_1 = v_1 = w'_1 = m'_{s1} = 0.$

Applying equation (5), these conditions can be put into the matrix form

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -3 & 0 \\ 0 & 0 & 0 & -3 \end{bmatrix} \begin{Bmatrix} u \\ v \\ w \\ m_s \end{Bmatrix}_1 + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 4 \end{bmatrix} \begin{Bmatrix} u \\ v \\ w \\ m_s \end{Bmatrix}_2 + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \begin{Bmatrix} u \\ v \\ w \\ m_s \end{Bmatrix}_3 = 0.$$

where the above 3 matrices are DL, DG, and DF within the SATANS programs.

For N= 1, $u_1 \pm v_1 = u' = w = m_s = 0,$

where the plus sign applies at an initial pole, and the minus sign at a final pole. The matrix form for these conditions is

$$\begin{bmatrix} -3 & 0 & 0 & 0 \\ 1 & \pm 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} u \\ v \\ w \\ m_s \end{Bmatrix}_1 + \begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} u \\ v \\ w \\ m_s \end{Bmatrix}_2 + \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} u \\ v \\ w \\ m_s \end{Bmatrix}_3 = 0.$$

For N=2, $u = v = w = m'_s = 0$

the matrix form is

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -3 \end{bmatrix} \begin{Bmatrix} u \\ v \\ w \\ m_s \end{Bmatrix}_1 + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 \end{bmatrix} \begin{Bmatrix} u \\ v \\ w \\ m_s \end{Bmatrix}_2 + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \begin{Bmatrix} u \\ v \\ w \\ m_s \end{Bmatrix}_3 = 0.$$

For N > 2, $u = v = w = m_s = 0$

and DL= identity matrix, DG= DF= null matrices.

The sclution procedure in SATANS is an elimination scheme and starts with

$$\{z\}_1 = -[P]_1\{z\}_2 + \{x\}_1 \quad , \tag{6}$$

where the values in $[P]_1$ based upon the Euler approximation are defined in Reference 14. The higher order approximation defines a new $[P]_1$. This new $[P]_1$ is obtained by simultaneously solving the pole conditions

$$[DL]\{z\}_1 + [DG]\{z\}_2 + [DF]\{z\}_3 = \{0\}, \tag{7}$$

and the domain equation at station 2 next to the pole

$$[C]_2\{z\}_1 + [B]_2\{z\}_2 + [A]_2\{z\}_3 = \{g\}_2 \quad , \tag{8}$$

to eliminate $\{z\}_3$. Thus,

$$\{z\}_3 = [A]_2^{-1}(\{g\}_2 - [C]_2\{z\}_1 - [B]_2\{z\}_2). \tag{9}$$

Substituting equation (9) into equation (7) gives

$$[DL]\{z\}_1 + [DG]\{z\}_2 + [DF][A]_2^{-1}(\{g\}_2 - [C]_2\{z\}_1 -$$
$$[B]_2\{z\}_2) = 0. \tag{10}$$

Combining like coefficients of the {z} vector leads to

$$([DL] - [DF][A]_2^{-1}[C]_2)\{z\}_1 + ([DG] - [DF][A]_2^{-1}[B]_2)$$
$$\{z\}_2 = -[DF][A]_2^{-1}\{g\}_2. \tag{11}$$

Finally, solving for $\{z\}_1$ yields

$$\{z\}_1 = -[DL - DF \times A_2^{-1} \times C_2]^{-1} [DG - DF \times A_2^{-1} \times B_2] \{z\}_2$$

$$+ [DL - DF \times A_2^{-1} \times C_2]^{-1} [- DF \times A_2^{-1}] \{g\}_2. \qquad (12)$$

Thus, $[P]_1 = -[DL - DF \times A_2^{-1} \times C_2]^{-1} [DG - DF \times A_2^{-1} \times B_2]$ and

$\{x\}_1 = [DL - DF \times A_2^{-1} \times C_2]^{-1} [- DF \times A_2^{-1}] \{g\}_2$. The new $[P]_1$ matrix has been placed into the "PMATRX" subroutine of SATANS-IIA and the new $\{x\}_1$ vector has been placed in the "FORCE" subroutine.

A listing of the pole routine may be found in Appendix D. To incorporate this new routine into a SATANS-I or-II program, first proceed to the "PMATRX" subroutine and remove the fifteen cards that are between, but not including, "IF(NN.GT.2) GO TO 90" and "11 CONTINUE". These cards are located after statement number "14" and just before statement number "11". Replace the cards removed by the ones listed in Appendix D which read from" C IN PMATRX " to " 90 M3=MN". Then proceed to the "FORCE" subroutine and remove statement number "10". Replace statement number "10" with the nine cards listed in Appendix D which read from " C IN FORCE" to "DO 11 I= 1,4" . Also place " COMMON /IBL5/IBCINL, IBCFNL " into the common area of the " FORCE " subroutine.

This completes the implementation of the new pole routine into either SATANS-I or II.

IV. PROBLEM DESCRIPTION

The geometry of the shallow spherical shell used in this study is identical to that used in Reference 1. Briefly, the shallow shell can be specified by the non-dimensional parameter λ , where

$$\lambda = 2[\ 3\ (\ 1 - \nu^2\ )\ ]^{1/4}\ (\ H/\ h\ )^{1/2} . \tag{1}$$

H is the rise of the shell, h is the thickness, and ν is Poisson's ratio. The mass density of the shell is m. All shells analyzed had the following dimensions;

Radii of Curvature	$R_s = R_\theta = 250$ inches
Thickness	h = 0.25 inches
Modulus of Elasticity	E = 30,000,000 psi
Poisson's Ratio	$\nu = 0.3$

All buckling pressures obtained will be listed as a percent of the classical buckling pressure of a complete sphere, q_0, where

$$q_0 = [\ 2\ E\ (\ h/R_s)^2\ ]\ /\ [\ 3\ (\ 1 - \nu^2\ )\ ]^{1/2} \tag{2}$$

Forty stations were used over the meridian. The nondimensional time increment δ t, where

$$t = T\ /\ (\ R_s^2 m\ /\ E)^{1/2} , \tag{3}$$

was taken as 0.05 for 3000 time steps, which is a total nondimensional time of 150. In addition, the axisymmetric analysis was repeated with a larger time step of $\delta t = 0.2$ for a total time of 600. In this study m was selected such that t is equal to T. The necessity for the long response time is explained in Reference 6.

In the axisymmetric analysis only the N= 0 harmonic is considered. However, in the asymmetric analysis a second harmonic is excited by applying an incremental load in that harmonic. In addition, analyses of the shells λ = 6, 7.5, and 11 were made using five harmonics. The step pressure load for the axisymmetric harmonic is

$$\{q^{(0)}\} = P\, q_0\ \{1\}, \tag{4}$$

and the step pressure load for the asymmetric second harmonic is

$$\{q^{(n)}\} = P\, q_0\ \varepsilon^{(n)}\ \{1\}, \tag{5}$$

where $n > 0$, and $\varepsilon^{(n)}$ is taken as 0.0001. The value taken for the second harmonic in the asymmetric analysis was the same as the critical harmonic for the static buckling analysis presented by Stilwell and Ball [2]. When there was an uncertainty as to which was the critical static harmonic the two harmonics in question were both tested. Run times using SATANS-IIA with a two-harmonic analysis for 3000 time steps and 40 stations on the meridian took an average of 28 minutes on the IBM 360/67.

The parameter used to determine the minimum load at which dynamic buckling occurs is the peak value of $\bar{V}$, called

$\bar{V}_{MAX}$, where $\bar{V}$ is defined as

$$\bar{V} = \int_0^{r_0} r\, W^{(0)}\, dr \Big/ \int_0^{r_0} r\, \xi\, dr \qquad (6)$$

r is the normal distance from the axis to the shell, r_0 is the maximum value of r, $W^{(0)}$ is the normal displacement of the axisymmetric response and ξ is the vertical distance from the base plane to the undeformed shell. The $\bar{V}$ is a measure of the volume of the shell deformation. The Fortran statements computing $\bar{V}$ and $\bar{V}_{MAX}$ are given in Appendix E. When working a problem that requires these calculations the nineteen cards are inserted directly into the "DYNAMIC" subroutine right after the "IF" statement that calls the "OUTPUT" subroutine.

For convenience, the response in each asymmetric harmonic is also measured using equation (6), with $W^{(0)}$ replaced with $W^{(n)}$. The parameter $\bar{V}$ for the asymmetric harmonics does not represent a volume of deformation as it does for the axisymmetric harmonic. It can, however, be used to indicate the relative excitation of the asymmetric harmonics.

The buckling criterion for both the axisymmetric and the asymmetric dynamic buckling analysis defines the critical load as that load P where a very small increase in P causes a very large increase in $\bar{V}_{MAX}$. This is the same criterion

as that used in Ref. [1].

V. RESULTS AND DISCUSSION

A. STATIC AXISYMMETRIC BUCKLING ANALYSIS

Table I presents the new results from the static axisymmetric buckling analyses for λ = 4 through 13 using the new pole routine. The two upper curves in Figure 1 present a comparison of the new results obtained by SATANS-IIA with those obtained by Stilwell and Ball [2] using the SATANS-I program. As can be seen in this figure, fairly significant changes in the buckling load occurred in the neighborhood of λ = 4,5, and 9; and somewhat smaller differences occurred in the region λ = 10 through 13. The upper data points in Figure 2 present the comparison of the new results from SATANS-IIA with those obtained by Huang [4]. This comparison shows a very good agreement between the two sets of results, except for the largest values of λ. The new results have eliminated the differences that existed between the SATANS-I results and Huang's results.

B. DYNAMIC AXISYMMETRIC BUCKLING ANALYSIS

Figure 3 presents the new results for the peak value of $\bar{V}_{MAX}$ versus P for the various values of λ tested. Table II presents all of the new results for the dynamic axisymmetric buckling load. These loads are selected from figures constructed just like Figure 3. In every case,

except for $\lambda = 4$, a value of P slightly above the P_{CRIT} value caused a $\bar{\bar{V}}_{MAX}$ indicative of buckling, as well as a nonconvergence of the iterative solution procedure.

The lower two curves of Figure 1 present a comparison versus λ of the new axisymmetric dynamic buckling results with the previous buckling results obtained by Ball and Burt [1]. In every case the new critical pressure is lower than the critical pressure obtained using the Euler approximation at the pole.

The lower data points of Figure 2 present a comparison of the new results with those obtained by Huang [5], by Stephens and Fulton [6], and by Stricklin [8]. Just as in the case of the static axisymmetric buckling analysis, the new results compare much more favorably with the other results than did the results of Reference 1. It's interesting to note that the new results now tend to be slightly lower than the other results, whereas the results of Reference 1 were higher for almost all values of λ .

C. DYNAMIC ASYMMETRIC BUCKLING ANALYSIS

Table III presents the new results for the critical pressures obtained from the dynamic asymmetric analysis. The second harmonics, or critical static harmonics, used in the analyses are also presented in Table III. A comparison of the critical pressures from the asymmetric analyses, Table III, with the critical pressures from the axisymmetric analyses, Table II, reveals that only the shell $\lambda = 6$ buckled at a load below the axisymmetric buckling load. For the shell $\lambda = 7$ the critical buckling load was slightly

larger when asymmetric motion was considered. In all other cases the buckling was not influenced by the presence of the second harmonic. These new buckling results and those by Ball and Burt [2] are plotted in Figure 4. The new results can be seen to be significantly different from the SATANS-I results, where the asymmetric buckling loads were lower than the axisymmetric loads for five out of the ten values of tested.

Except for λ = 6 and 7, the relationship between $\bar{V}_{MAX}$ and P for the N= 0 harmonic, in the two-harmonic analyses, was found to be essentially identical to the relationship found in the axisymmetric buckling analysis shown in Figure 3. Table IV A presents the $\bar{V}_{MAX}$ versus P data for both the N= 0 harmonic and the second harmonic, for all values of λ tested, except for λ = 6. Note that, except for λ =7 and 11, $\bar{V}_{MAX}$ for the asymmetric harmonic is generally very small, even when the $\bar{V}_{MAX}$ for the N= 0 harmonic indicates that the shell has buckled. Thus, except for the shells λ = 6 and 7, the presence of the asymmetric motion does not influence the axisymmetric motion, and except for the shells λ = 6, 7 and 11 the asymmetric motion is very small prior to buckling in the axisymmetric harmonic.

A more detailed analysis of the shell λ = 6 has been conducted since it was the only shell that revealed any significant axisymmetric sensitivity to asymmetric motion. This shell was studied using two two-harmonic analyses (N= 0, 1 and N= 0, 2) and a five-harmonic analysis (N= 0, 1, 2, 3, and 4). Figure 5 and Tables IV B and IV C contain values of $\bar{V}_{MAX}$ versus P for both of the asymmetric harmonics, N= 1

and N= 2, in the two two-harmonic analyses, as well as the values of $\bar{V}_{MAX}$ for the axisymmetric harmonic, N= 0. Figure 6 and Table IV D present the values of $\bar{V}_{MAX}$ versus P for the N= 0,1,2,3, and 4 harmonics from the five-harmonic study. A comparison of the critical buckling load predicted from the results of the two two-harmonic analyses in Figure 5 with the critical load from the five-harmonic analysis obtained from Figure 6 shows that the presence of the additional harmonics results in the shell buckling at a slightly lower load (0.50), with significant motion in the N= 1 harmonic instead of the N= 2 harmonic (see the nonconverged solution at P= 0.51), which is the critical harmonic for static asymmetric buckling. Studies using five harmonics have also been conducted for λ = 7.5 and λ = 11. As can be seen in Table IV D the critical harmonic for λ = 7.5 remained N= 3; however, significant motion occurred in that harmonic at P= .41 and .44. In the case of λ = 11, relatively large asymmetric motion occurred in the asymmetric mode of N= 5 vice 6 at a value of P= .46.

The comparison of the new results for the critical pressure for dynamic asymmetric buckling with those obtained analytically by Stricklin [8], by Akkas [9], and experimentally by Lock et al [7] is illustrated in Figure 7. The comparison reveals an agreement with Stricklin in every case, in general a higher value of P_{CRIT} than those obtained by Akkas, and most importantly a very good agreement with Lock's experimental results.

When making the comparison between the new results and those obtained by Akkas, it is necessary to look at the differences in the problem solution parameters used in the two studies. For example, buckling results obtained from SATANS-IIA using the same time increment as used by Akkas,

$\delta t = .2$ for 3000 time steps, were significantly higher than those using the time step of $\delta t = .05$ for many values of λ. Furthermore, the new results had , in some cases, instances of buckling occurring as far out in time as 130. Akkas, to shorten computer run times, observed the cap only for a time of less than 5. Furthermore, only the harmonics N= 1 or 2 or 3 were studied by Akkas for shells λ = 5 through 12. If the critical harmonic is not studied, the predicted load will be too high. Thus, it appears that Akkas' lower bound loads may not be true lower bounds.

Two additional features of the shell response should be noted. First, shells λ = 6, 7.5, and 11 exhibited a non-buckled response in the axisymmetric harmonic to a load larger than the defined critical buckling load. This can be seen in Tables IV A and IV C. Second, and most importantly, the buckling load proposed by Ball and Burt [1], and used here, defines buckling to occur when the $\bar{V}_{MAX}$ in the axisymmetric harmonic undergoes a large change due to a small change in P. Another criterion for dynamic buckling in the asymmetric analysis discussed in Reference 1 is to define the buckling load as that threshold load that initiates significant growth in the asymmetric harmonic. Re-examination of the $\bar{V}_{MAX}$ versus P data in Table IV A through D reveals that shells λ = 6, 7, and 11 exhibited relatively large asymmetric motion at loads smaller that the defined buckling load when compared with other $\bar{V}_{MAX}$ values for those shells, even though the numbers themselves were small when compared with the axisymmetric harmonic. Shells λ =7.5 and 12 appear to be borderline cases. If the alternate criterion for buckling is used, the critical buckling loads for shells λ = 6, 7, and 11 become 0.47, 0.45, and 0.45, respectively. The shells λ = 7.5 and 12 could have buckling

loads as low as 0.40 and 0.44, respectively. These values are more conservative than the definition based upon axisymmetric response. These five shells are the same five shells that exhibited an asymmetric buckling load lower than the axisymmetric buckling load in Reference 1.

VI. SUMMARY AND CONCLUSIONS

A digital computer program for the geometrically nonlinear analysis of totally arbitrarily loaded shells of revolution (SATANS-II) was modified to more accurately account for the conditions at the pole of the shell. This program, called SATANS-IIA, was used to determine the buckling load of shallow spherical shells of various sizes when subjected to static axisymmetric, dynamic axisymmetric, and dynamic nearly axisymmetric step-pressure loads of infinite duration. The cap sizes ranged from $\lambda = 4$ to 13 including $\lambda = 7.5$. A comparison was made between the new buckling results with the improved pole handling routine and the results that did not have the new pole routine. The comparison revealed a significant change in buckling pressures, due solely to the change from an order Δ finite difference approximation of the first derivatives at the pole to an approximation of order Δ^2. These new critical pressures are in very good agreement with the results from other studies of the same spherical shells. This good agreement with other results, which came about as a result of the modification of the pole handling routine, is a strong indication that the manner in which the pole condition is handled is vital to the accuracy of the solutions obtained.

In the asymmetric analysis, two harmonics were included for most of the shells; the axisymmetric harmonic and one asymmetric harmonic. Five-harmonic analyses were conducted for three of the shells. Two buckling criteria for the

asymmetric analysis were considered. One defined buckling as that threshold load that caused a large increase in a deformation parameter, $\bar{V}_{MAX}$, in the axisymmetric harmonic. The other, more conservative than the first, defined buckling as that threshold load that caused a large increase in the $\bar{V}_{MAX}$ value for the asymmetric harmonic. Both values have been presented.

The new static axisymmetric, dynamic axisymmetric, and even the dynamic asymmetric critical buckling pressure loads appear to be fairly reliable results for perfect, shallow shells. The effect of realistic imperfections remains to be determined.

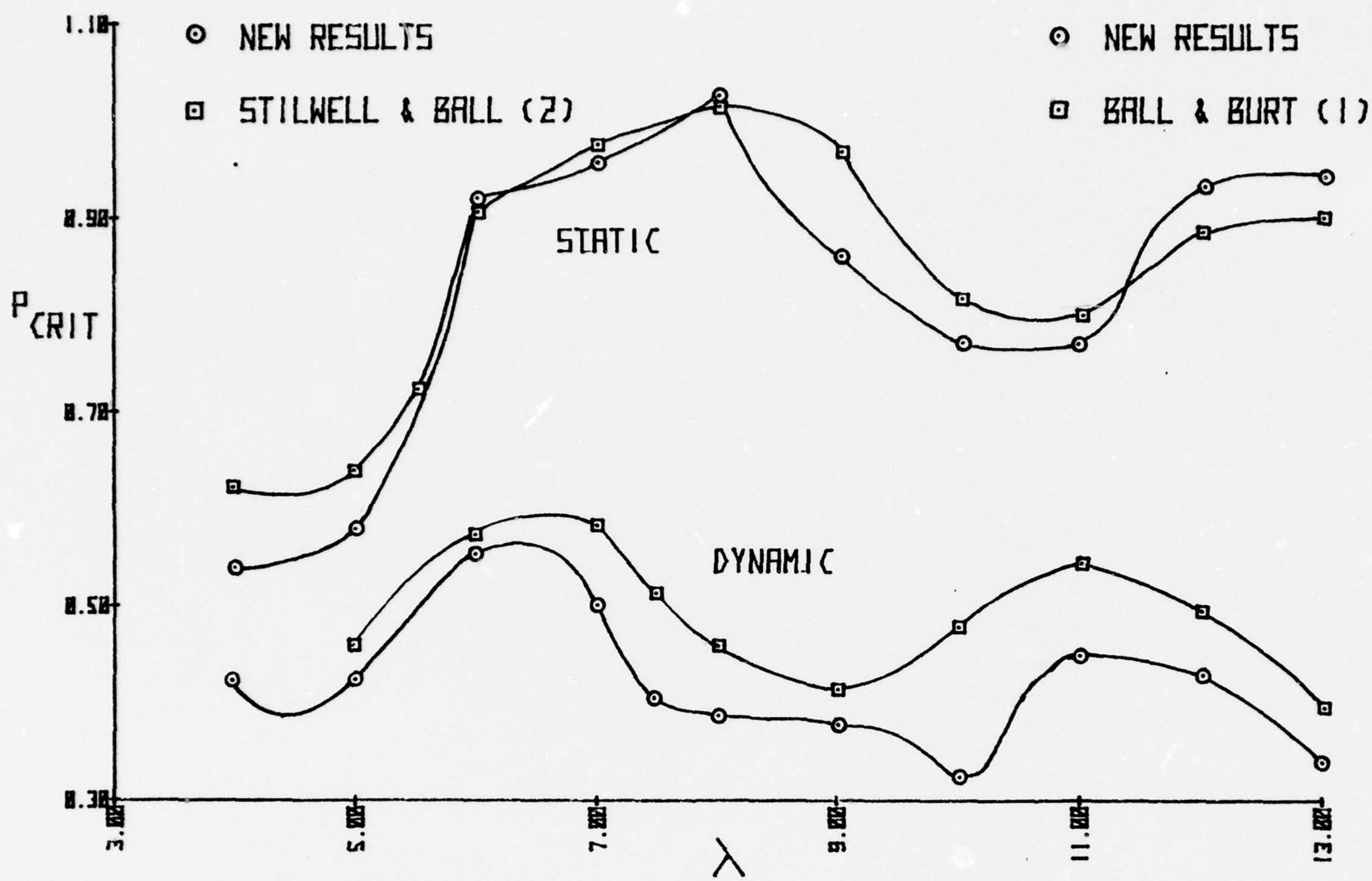


Figure 1 - CRITICAL STEP-PRESSURE LOAD VERSUS λ AXISYMMETRIC (SATANS-I VERSUS SATANS-IIA)

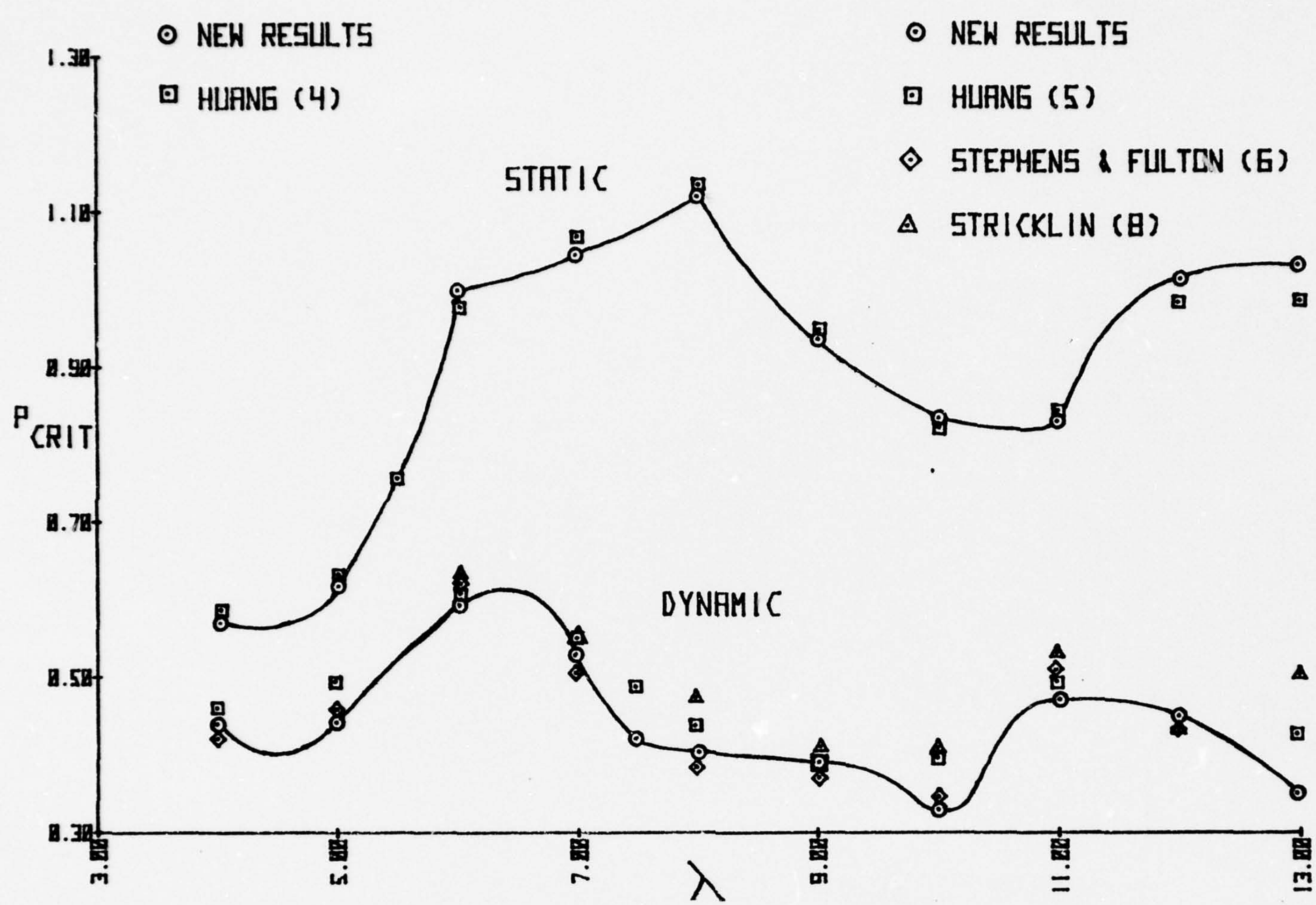


Figure 2 - CRITICAL STEP-PRESSURE LOAD VERSUS λ
AXISYMMETRIC (SATANS-IIA VERSUS ALL OTHERS)

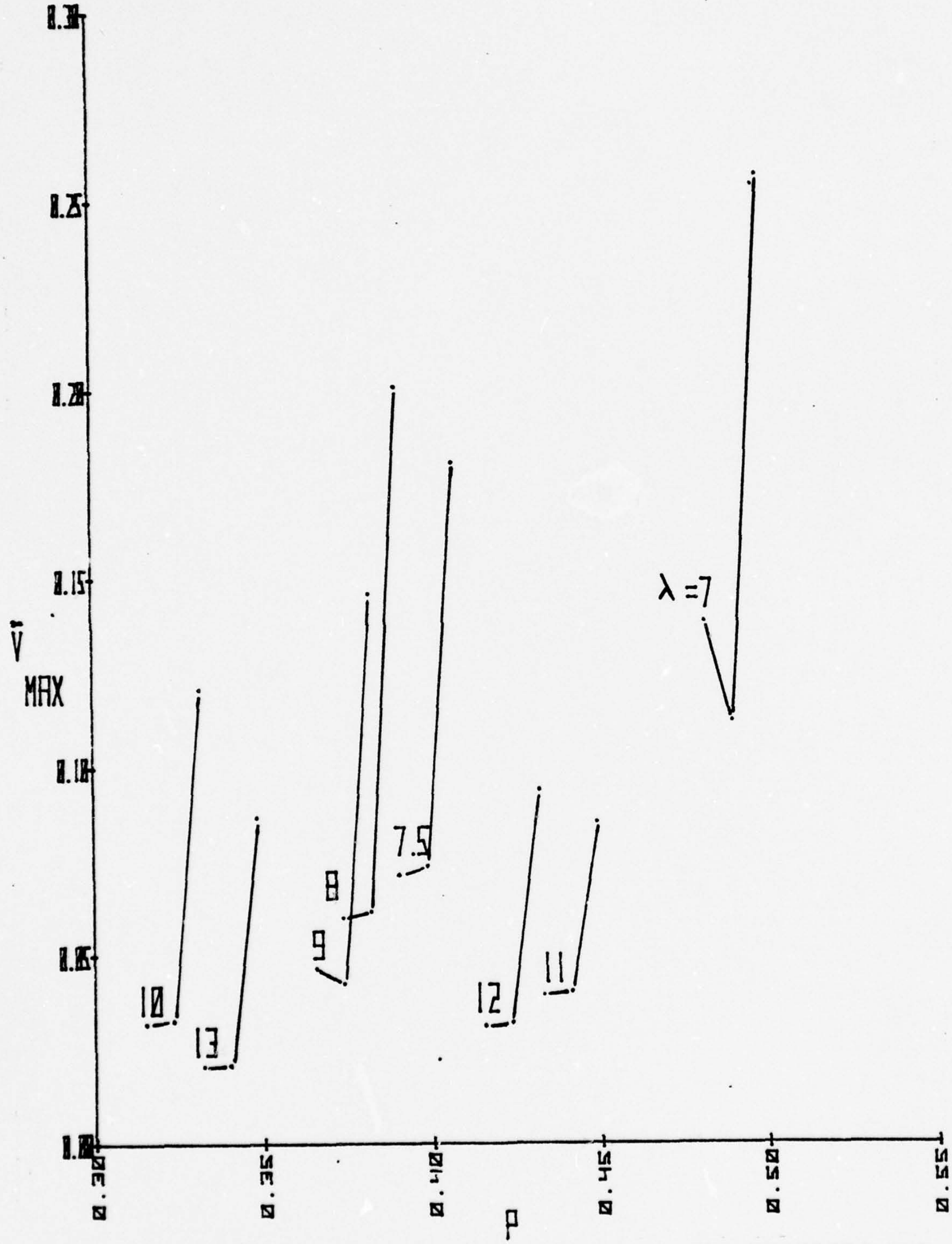


Figure 3 - PEAK DEFLECTION VERSUS P, AXISYMMETRIC AND ASYMMETRIC CASES FOR VARIOUS VALUES OF λ (SATANS-IIA)

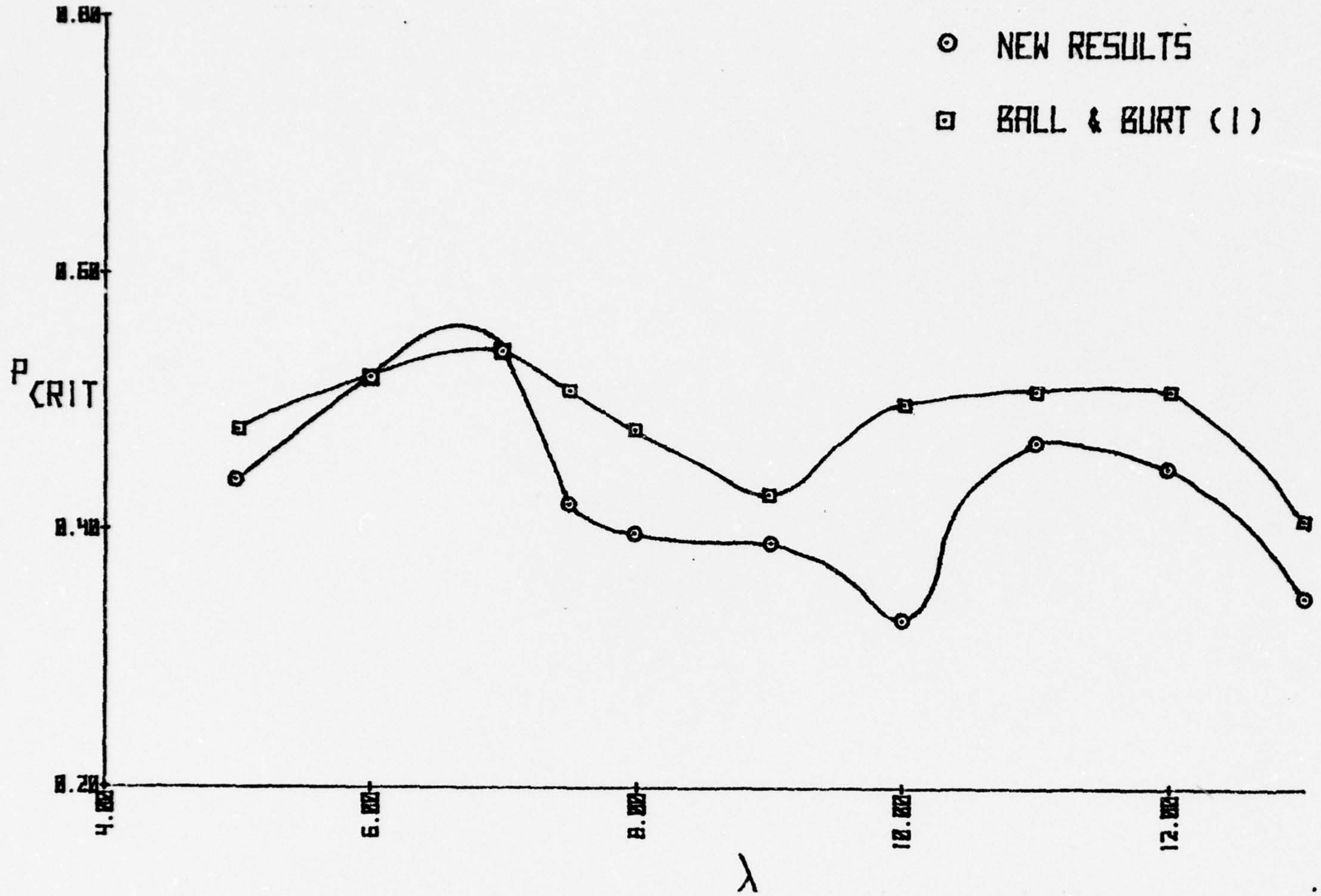


Figure 4 - CRITICAL STEP-PRESSURE LOAD VERSUS λ
ASYMMETRIC ANALYSES (SATANS-I VERSUS SATANS-IIA)

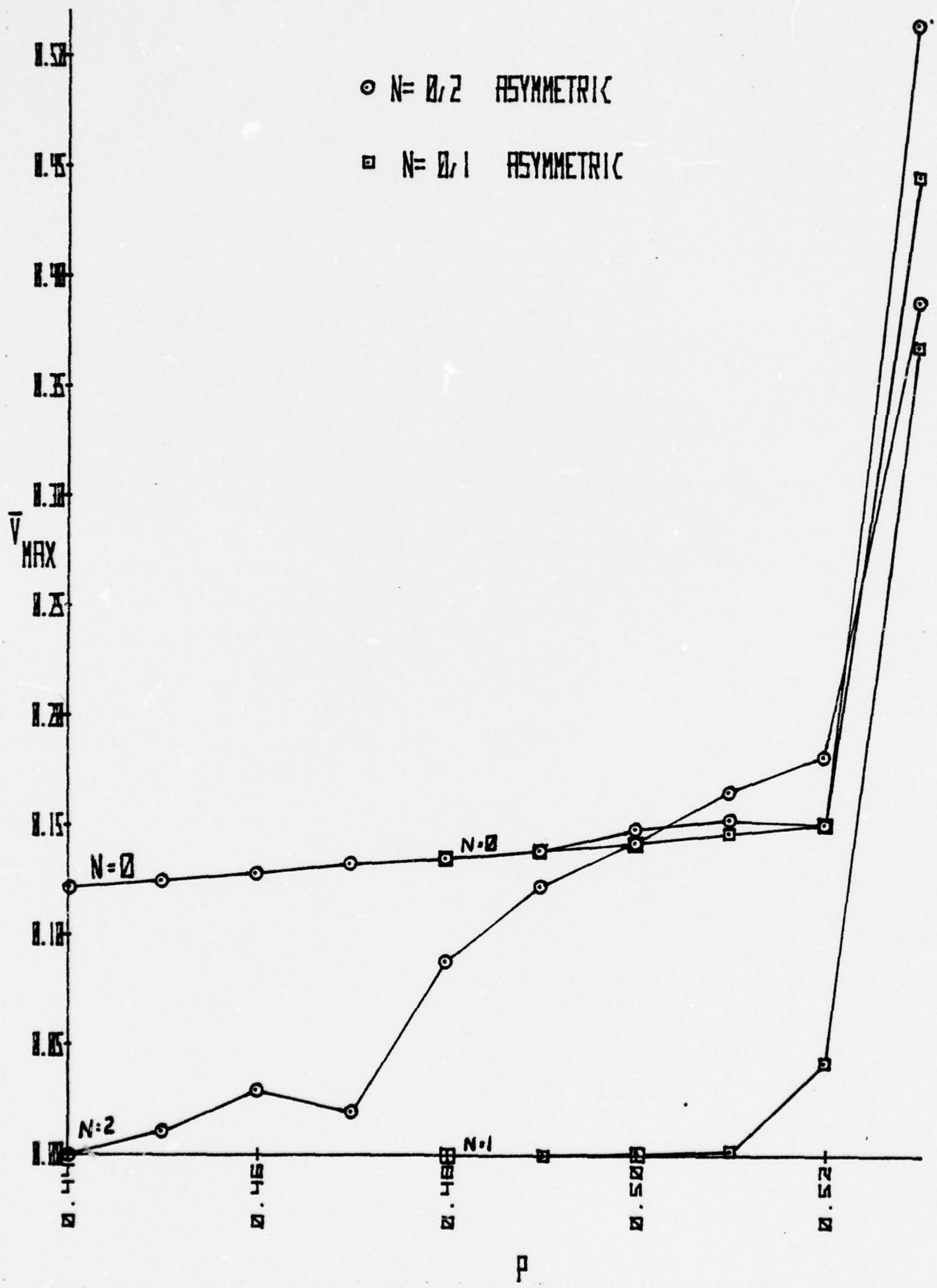


Figure 5 - PEAK DEFLECTION VERSUS P FOR THE ASYMMETRIC ANALYSES OF λ = 6 (N=0,1 AND N=0,2)

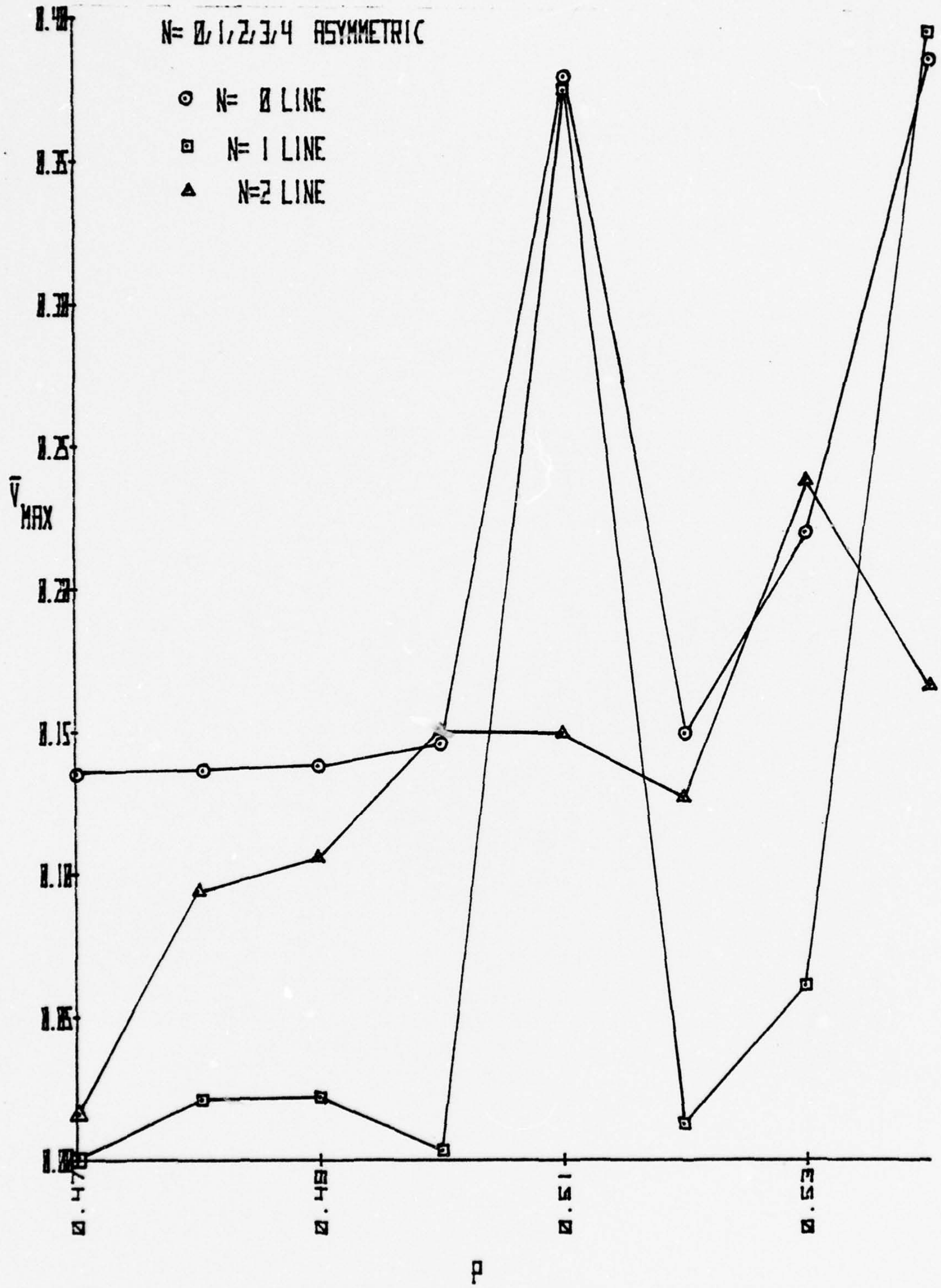


Figure 6 - PEAK DEFLECTION VERSUS P FOR THE ASYMMETRIC ANALYSES OF λ = 6 (N=0,1,2,3,AND4, ONLY N=0,1,AND2 PLOTTED)

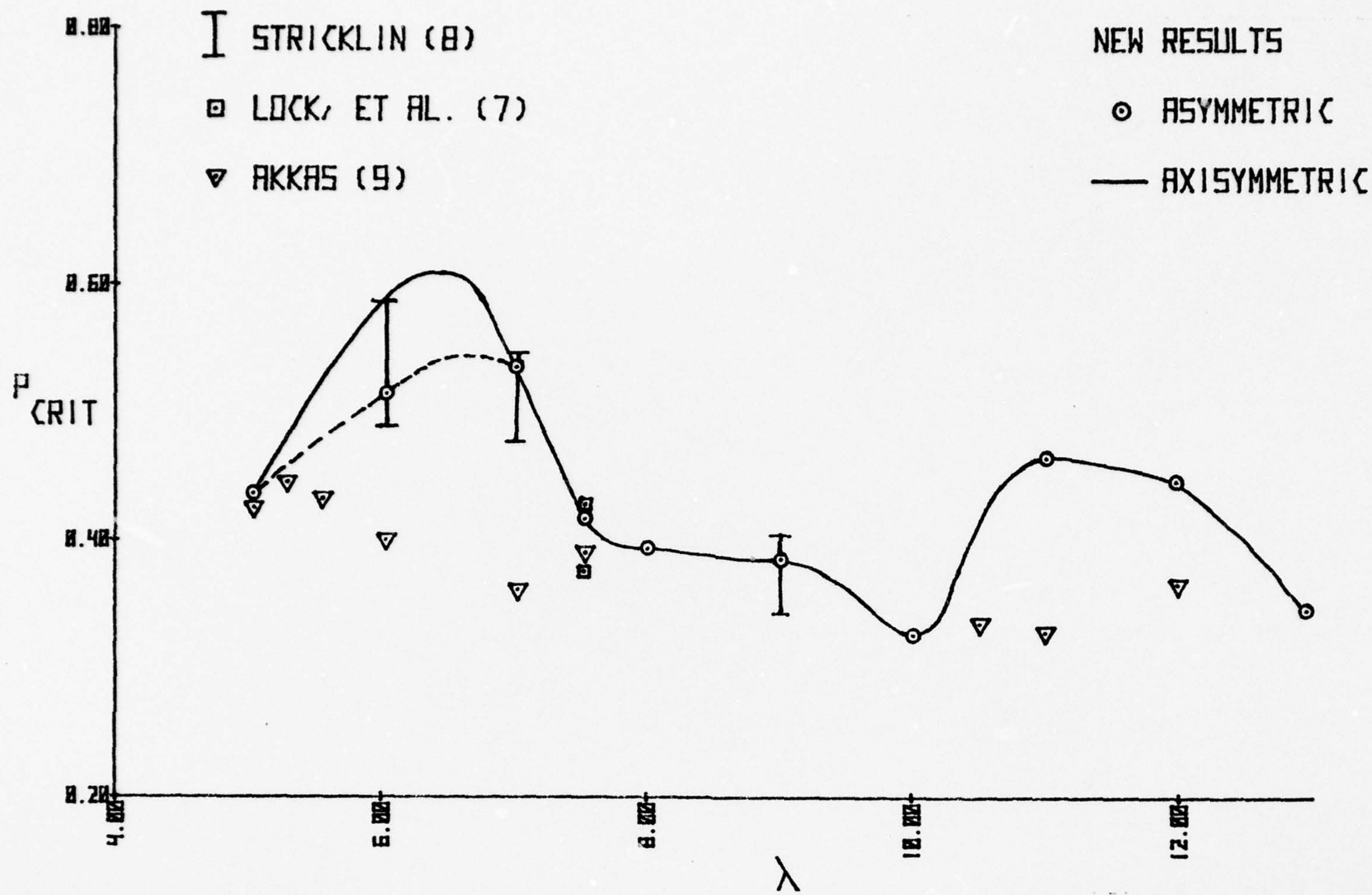


Figure 7 - CRITICAL STEP-PRESSURE LOAD VERSUS λ
ASYMMETRIC ANALYSES (SATANS-IIA VERSUS ALL OTHERS)

A. TABLES

1. TABLE I Critical pressure loads from the static axisymmetric analyses.

λ	4	5	6	7	8	9	10	11	12	13
P_{CRIT}	.568	.616	1.0	1.048	1.12	.936	.832	.832	1.016	1.032

2. TABLE II Critical step-pressure loads from the axisymmetric dynamic analyses.

λ	4	5	6	7	7.5	8	9	10	11	12	13
P_{CRIT}	.45	.44	.59	.53	.42	.40	.39	.33	.47	.45	.35

3. TABLE III Critical step-pressure loads from the dynamic asymmetric analyses and critical asymmetric harmonics.

λ	5	6	7	7.5	8	9	10	11	12	13
P_{CRIT}	.44	.52	.54	.42	.40	.39	.33	.47	.45	.35
N_{CRIT}	1	2	3	3	4	5	6	7	8	9

4. TABLE IV Dynamic asymmetric analyses for $\bar{V}_{MAX}$ versus P.

1. TABLE IV A. Two-harmonic analyses for all values of λ exept $\lambda = 6$.

λ = 5 N= 0 and 2

P	.43	.44	.45
N= 0	.1659	.1676	.6606
N= 2	.0004787	.0000566	.0687

N= 0 and 1

P	.44	.45
N= 0	.1675	.6606
N= 1	.0003145	.0687

P	.46
N= 0	.7653
N= 1	.001092

λ = 7, N= 0 and 3

P	.45	.46	.47	.48	.49	.50	.52
N= 0	.09452	.09571	.09812	.1005	.1029	.1052	.1099
N= 3	.000889	.007456	.05052	.04323	.0279	.0335	.07488

P	.53	.54	.55
N= 0	.1122	.1146	.2709
N= 3	.05997	.06252	.03809

λ = 7.5, N= 0 and 3

P	.40	.41	.42	.43	.44	.45
N= 0	.0703	.07228	.07429	.2636	.07837	.2076
N= 3	.0001094	.001296	.0004304	.002754	.001188	.000338

P	.46
N= 0	.200
N= 3	.0003276

λ = 8, N= 0 and 4

P	.38	.39	.40	.41	.42	.43
N= 0	.05893	.0607	.0624	.1964	.1713	.1957
N= 4	.0000566	.0000703	.0000364	.0000333	.0000274	.0000299

P	.44
N= 0	.2297
N= 4	.0000326

λ = 9 N= 0 and 4

P	.38	.39	.40
N= 0	.04738	.04875	.1576
N= 4	.00003597	.00004635	.00004497

N= 0 and 5

P	.40
N= 0	.05012
N= 5	.00008385

λ = 10, N= 0 and 5

P	.32	.33	.34	.36	.38	.40
N= 0	.03239	.03347	.1086	.1217	.1288	.1235
N= 5	.0000281	.0000472	.00004125	.00002103	.0000449	.000114

λ = 11, N= 0 and 6

P	.45	.46	.46	.48	.49	.50
N= 0	.03910	.04004	.04099	.09814	.04241	.08824
N= 6	.004595	.01332	.02232	.02864	.03955	.02813

λ = 12, N= 0 and 7

P	.44	.45	.46
N= 0	.03236	.03316	.08633
N= 7	.00004214	.0004561	.00005158

λ = 13, N= 0 and 8

P	.34	.35	.36	.38	.40
N= 0	.02119	.02185	.06637	.07844	.07381
N= 8	.00001148	.00001134	.000006607	.000008245	.000119

2. TABLE IV B. Two-harmonic analyses with N= 0 and 1, λ = 6 cnly.

P	.48	.49	.50	.51	.52	.53
N= 0	.1350	.1385	.1421	.1460	.1499	.4453
N= 1	.0002797	.000195	.000245	.000926	.04081	.3668

3. TABLE IV C. Two-harmonic analyses with N= 0 and 2, λ = 6 only.

P	.44	.45	.46	.47	.48	.49	.50
N= 0	.1218	.1250	.1276	.1320	.1350	.1385	.1479
N= 2	.000239	.0101	.0293	.01976	.08768	.1223	.1419

P	.51	.52	.53	.54	.55	.56
N= 0	.1526	.1499	.5137	.5060	.2040	.5305
N= 2	.1654	.1816	.3878	.3996	.2156	.3617

4. TABLE IV D Five-harmonic analyses for selected shells.

λ = 6 N= 0,1,2,3, and 4

P	.47	.48	.49	.50	.51	.52	.53
N= 0	.1313	.1347	.1382	.1460	.3797	.1498	.2200
N= 1	.00021	.02108	.02215	.003676	.3743	.01276	.0616
N= 2	.0187	.0953	.1069	.1507	.1502	.1279	.2385
N= 3	.000181	.006237	.01437	.00163	.0405	.0123	.03978
N= 4	.0031	.04757	.05428	.04402	.05896	.0495	.064

P	.54
N= 0	.3854
N= 1	.3953
N= 2	.1671
N= 3	.05298
N= 4	.0613

λ = 7.5 N= 0,1,2,3,4 and 4

P	.40	.41	.42	.43	.44	.45
N= 0	.0703	.07228	.07429	.2592	.07837	.2544
N= 1	.00004855	.00004198	.00006093	.0002737	.0001167	.005952
N= 2	.0001164	.00007456	.0004184	.0000982	.000788	.0003188
N= 3	.0001277	.001187	.0004597	.0002853	.00107	.0003188
N= 4	.0008224	.0001898	.0002448	.0000526	.000280	.000134

λ = 11 N= 0,4,5,6, and 7

P	.45	.46	.47	.48	.49	.50
N= 0	.03910	.04004	.0499	.04195	.04291	.1040
N= 4	.0005759	.001263	.0009774	.001388	.002657	.001565
N= 5	.009568	.0140	.0124	.02239	.02759	.01548
N= 6	.002560	.007828	.02330	.02767	.02602	.02644
N= 7	.0001743	.0001486	.0002021	.01202	.02048	.02064

APPENDIX A

LISTING OF SATANS-IIA

```
C*************************************************************************SAT00010
C      THIS PROGRAM SERVES AS THE 'MAIN' PROGRAM, AND CALLS 'SATANS',    *SAT00020
C      AND CHECKS 'IRNAGN' TO SEE IF WE DESIRE ANOTHER RUN                *
C*************************************************************************SAT00030
      DIMENSION P(4,4,124),DEE(4,4,124),DST(4,4,124),X(4,124),Z(4,132),
     1ZO(4,132),Z2(4,132),Z3(4,132),ZDOT(4,132),IS(99,4),JS(99,4),
     2ID(99,4),JD(99,4),PHIXB(124),PHITB(124)
      COMMON /BLRUN/ IRNAGN
      COMMON /BLTHTA/ THETAM,COEFF
    1 CONTINUE
      CALL SATANS (P,DEE,DST,X,Z,ZO,Z2,Z3,ZDOT,IS,JS,ID,JD,PHIXB,PHITB)
      IF (IRNAGN.EQ.1) GO TO 1
      CALL FLYNVY                                                         SAT00080
      STOP                                                                SAT00090
      END                                                                 SAT00100
                                                                          SAT00110
      SUBROUTINE GEOM                                                     SAT00120
C*************************************************************************SAT00130
C         THIS SUBROUTINE COMPUTES THE NONDIMENSIONAL GEOMETRY FUNCTIONS*SAT00140
C         OF THE SHELL                                                  *SAT00150
C*************************************************************************SAT00160
      REAL NU,LAM,LAM2,JAY,MT,LSD18,LSD1N,MASS                            SAT00170
      COMMON /IBL4/ KMAX,KL                                               SAT00180
      COMMON /BL8/  R(500),GAM(500),OMT(500)                              SAT00190
      COMMON /BL11/ OMXI(500),PHEE,TO,T2                                  SAT00200
      COMMON /BL17/ DEL                                                   SAT00210
      COMMON /BL20/ DEOMX(500)                                            SAT00220
      COMMON /BL32/ TKN,ELAST,CHAR,SIGC                                   SAT00230
      COMMON /BL102/ DELOAD                                               SAT00240
      COMMON /BL103/ MASS(500)                                            SAT00250
      COMMON /BLTHTA/ THETAM
      AKX=KMAX-1
      DEL=1./AKX
      THET=ARSIN(2.2801/CHAR)
      DO 11 K=1,KMAX
      AK=K
      R(K)=(7.9499+(AK-1.)*(2.2801)/AKX)/CHAR
      GAM(K)=(2.2801/CHAR)/R(K)
      OMXI(K)=0.
      DEOMX(K)=0.
      OMT(K)=COS(THET)/R(K)
      MASS(K)=1.
   11 CONTINUE
      RETURN                                                              SAT00350
      END                                                                 SAT00360
                                                                          SAT00370
      SUBROUTINE BDB(K,B,DB,D,DD)                                         SAT00380
C*************************************************************************SAT00390
```

```
C         THIS SUBROUTINE COMPUTES THE NONDIMENSIONAL IN-PLANE AND        *SAT00400
C         BENDING STIFFNESSES OF THE SHELL                                *SAT00410
C*************************************************************************SAT00420
      REAL NU,LAM,LAM2,JAY,MT,LSD18,LSD1N                                  SAT00430
      COMMON /BL15/ NU,U1(99),V1(99),W1(99),V2(99),U2(99),W2(99),U3(99),   SAT00440
     1             V3(99),W3(99)                                           SAT00450
      COMMON /BL17/ DEL                                                    SAT00460
      COMMON /BL32/ TKN,ELAST,CHAR,SIGC                                    SAT00470
      B=1.089082
      D=.09075683
      DE=0.
      DD=0.
      RETURN                                                               SAT00520
      END                                                                  SAT00530
                                                                           SAT00540
      SUBROUTINE PLOAD(K,Z)                                                SAT00550
C*************************************************************************SAT00560
C         THIS SUBROUTINE ESTABLISHES THE NON-DIMENSIONAL FOURIER         *SAT00570
C         COEFFICIENTS OF THE LOADS APPLIED TO THE SHELL                  *SAT00580
C*************************************************************************SAT00590
      REAL MASS                                                            SAT00600
      DIMENSION Z(4,1)                                                     SAT00610
      COMMON /IBL1/ MNMAX                                                  SAT00620
      COMMON /IBL2/NN(99),MNINIT                                           SAT00630
      COMMON /IBL4/ KMAX,KL                                                SAT00640
      COMMON /IBL8/ LSTEP,ITR                                              SAT00650
      COMMON /BL3/  PR(99),PX(99),PT(99)                                   SAT00660
      COMMON /BL6/  SOE,OSE,ALOAD                                          SAT00670
      COMMON /BL8/  R(500),GAM(500),OMT(500)                               SAT00680
      COMMON /BL32/ TKN,ELAST,CHAR,SIGC                                    SAT00690
      COMMON /BL102/ DELOAD                                                SAT00700
      COMMON /BL103/ MASS(500)                                             SAT00710
      COMMON /BL17/DEL/BL100/TEED,$DYNMC                                   SAT00720
      COMMON /BLTHTA/ THETAM,COEFF
      RETURN                                                               SAT00910
      END                                                                  SAT00920
      SUBROUTINE  INITL (Z,ZD,Z2,Z3,ZDOT)                                  SAT01110
C*************************************************************************SAT01120
C     THIS SUBROUTINE DESCRIBES THE INITIAL CONDITIONS FOR DYNAMIC CASESSAT01130
C*************************************************************************SAT01140
      IMPLICIT LOGICAL*1 ($)                                               SAT01150
      DIMENSION Z(4,1),ZD(4,1),Z2(4,1),Z3(4,1),ZDOT(4,1)                   SAT01160
      COMMON /IBL1/ MNMAX                                                  SAT01170
      COMMON /IBL2/ NN(99),MNINIT                                          SAT01180
      COMMON /IBL4/ KMAX,KL                                                SAT01190
      COMMON /IBL5/ MAXM                                                   SAT01200
      COMMON /IBL12/ KMAX1,KMAX2,NCONV                                     SAT01210
      COMMON /BL6/  SOE,OSE,ALOAD                                          SAT01220
```

```
      COMMON /BL32/ TKN,ELAST,CHAR,SIGO                                 SAT01230
      COMMON /BL100/ TEEO,$DYNMC                                        SAT01240
      COMMON /BL101/  DELSD                                             SAT01250
      NN(1)=0
      NN(2)=1
      NN(3)=2
      NN(4)=4
      PI=3.14159
      DO 2 M=1,MAXM
      IF(M.EQ.1) VEL=-444.08/PI
      IF(M.EQ.2) VEL=-444.08/2
      IF(M.EQ.3) VEL=-444.08*2./(3.*PI)
      IF(M.EQ.4) VEL=-444.08*2./(15.*PI)
      IF(M.EQ.4) VEL= 444.08*2./(15.*PI)
      DO 2 K=2,KL
      I=K+1+(M-1)*KMAX2
    2 ZOOT(3,I)=VEL*ELAST*TEEO/(CHAR*SIGO)*10
      RETURN                                                            SAT01260
      END                                                               SAT01270
      SUBROUTINE TLOAD(K,Z)                                             SAT00940
C***********************************************************************SAT00950
C     THIS SUBROUTINE DESCRIBES THE THERMAL LOADING ON THE SHELL       *SAT00960
C***********************************************************************SAT00970
      REAL NU                                                           SAT00980
      DIMENSION Z(4,1)                                                  SAT00990
      COMMON /IBL1/ MNMAX                                               SAT01000
      COMMON /IBL2/ NN(99),MNINIT                                       SAT01010
      COMMON /BL15/ NU,U1(99),V1(99),W1(99),V2(99),U2(99),W2(99),U3(99),SAT01020
     1              V3(99),W3(99)                                       SAT01030
      COMMON /BL32/ TKN,ELAST,CHAR,SIGO                                 SAT01040
      COMMON /IBL8/ LSTEP,ITR                                           SAT01050
      COMMON /BL5/  TT(99),EMT(99),DT(99),DMT(99)                       SAT01060
      COMMON /BL6/  SOE,OSE,ALOAD                                       SAT01070
      RETURN                                                            SAT01080
      END                                                               SAT01090
                                                                        SAT01100
      SUBROUTINE IMPERF (PHIXB,PHITB)                                   SAT01280
      DIMENSION PHIXB(1),PHITB(1)                                       SAT01290
      COMMON /IBL4/ KMAX,KL
      COMMON /IBL9/ MAXM
      KANDM=KMAX*MAXM
      DO 1 I=1,KANDM
      PHIXB(I)=0.
      PHITB(I)=0.
    1 CONTINUE
      RETURN                                                            SAT01300
      END                                                               SAT01310
      SUBROUTINE SATANS (P,CEE,DST,X,Z,ZC,Z2,Z3,ZOOT,IS,JS,ID,JD,PHIXB, SAT01320
```

```
     1PHITB)
C*********************************************************************
C     THIS SUBROUTINE READS ALL DATA, PRINTS THE OUTPUT TITLE PAGE,  *
C     AND PASSES CONTROL TO ONE OF THE MAJOR CONTROLLING SUBROUTINES *
C*********************************************************************
      IMPLICIT LOGICAL*1 ($)
      REAL*4 NU,LAM,LAM2,JAY,MT,LSD18,LSD1N,MASS
      DIMENSION P(4,4,1),DEE(4,4,1),DST(4,4,1),X(4,1),Z(4,1),
     1ZC(4,1),Z2(4,1),Z3(4,1),ZDOT(4,1),IS(99,1),JS(99,1),IC(99,1),
     2JC(99,1),PHIXB(1),PHITB(1)
      COMMON /IBL1/ MNMAX
      COMMON /IBL4/ KMAX,KL
      COMMON /IBL5/ IBCINL,IBCFNL
      COMMON /IBL9/ MAXM
      COMMON /IBL10/ IFREQ,NTHMAX
      COMMON /IBL13/ ITRMAX,LSMAX
     1/IBLJ/ JUMP
      COMMON /BL13/ OMEG1(4,4),CAPL1(4,4),OMEGL(4,4),CAPLL(4,4),
     1              UNIT(4,4)
      COMMON /BL15/ NU,U1(99),V1(99),W1(99),V2(99),U2(99),W2(99),U3(99),
     1              V3(99),W3(99)
      COMMON /BL16/ EPS
      COMMON /BL18/ EL1(4),ELL(4)
      COMMON /BL19/ TH(36)
      COMMON /BL32/ TKN,ELAST,CHAR,SIGC
      COMMON /BL100/ TEEC,$DYNMC
      COMMON /BL102/ DELOAD
      COMMON /BLPLOT/ IRADII,IGAMMA,IOMEGS,IOMEGT,IDEOMS,IBSTIF,IDSTIF,
     1                IBBSTF,IDDSTF,IPR,IPS,IPT,ITT,IMT,IDTT,IDMT,INS,
     2                INTH,INSTH,IQS,IMS,IMTH,IMSTH,IU,IV,IW,IPHIS,
     3                IPHIT,IPHI,$PLOTS,$MODAL
      COMMON /BLDATA/ TITLE,NO,IMODE,NDIMEN,IPRINT,LCHMAX,IC
      COMMON /BLRUN/ IRNAGN
      DIMENSION TITLE (18)
C*********************************************************************
C     READ IN DATA FOR THIS RUN                                      *
C*********************************************************************
      READ (5,100) TITLE
      READ (5,101) NO,$DYNMC,IMODE,NDIMEN,NTHMAX,IFREQ,IPRINT,IBCINL,
     1             IBCFNL,KMAX,MNMAX,MAXM,LSMAX,LCHMAX,ITRMAX,IC
      READ (5,102) NU,SIGO,ELAST,TKN,CHAR,TEEO
      READ (5,103) DELOAD,EPS
      READ(5,101) JUMP
      IF (NTHMAX.EQ.0) GO TO 1
      READ (5,104) (TH(NTH),NTH=1,NTHMAX)
      DO 2 NTH=1,NTHMAX
    2 TH(NTH)=TH(NTH)*6.2831853/360.0
    1 IF (IBCINL.EQ.0) READ (5,105) OMEG1,CAPL1,EL1
```

```
      IF (IBCFNL.EQ.0) READ (5,105) OMEGL,CAPLL,ELL                     SAT01810
      READ (5,106) $PLOTS,$MODAL,IRADII,IGAMMA,IOMEGS,ICMEGT,ICECMS,    SAT01820
     1             IBSTIF,IDSTIF,IBBSTF,IDDSTF,IPR,IPS,IPT,ITT,IMT,IDTT,SAT01830
     2             IDMT,INS,INTH,INSTH,IQS,IMS,IMTH,IMSTH,IU,IV,IW,     SAT01840
     3             IPHIS,IPHIT,IPHI                                     SAT01850
      READ (5,107) IRNAGN                                               SAT01860
C***********************************************************************SAT01870
C     THIS CONCLUDES DATA READ-IN FOR THIS RUN.                        *SAT01880
C     PRINT OUT PERTINENT DATA.                                        *SAT01890
C***********************************************************************SAT01900
      WRITE (6,200) NO                                                  SAT01910
      WRITE (6,201) TITLE                                               SAT01920
      WRITE (6,202)                                                     SAT01930
      IF (IBCINL.LT.0) WRITE (6,203)                                    SAT01940
      IF (IBCINL.LT.0) GO TO 3                                          SAT01950
      WRITE (6,204)                                                     SAT01960
      WRITE (6,205) (OMEGI(1,J),J=1,4),(CAPLI(1,J),J=1,4),ELI(1)        SAT01970
      WRITE (6,206) (OMEGI(2,J),J=1,4),(CAPLI(2,J),J=1,4),ELI(2)        SAT01980
      WRITE (6,207) (OMEGI(3,J),J=1,4),(CAPLI(3,J),J=1,4),ELI(3)        SAT01990
      WRITE (6,208) (OMEGI(4,J),J=1,4),(CAPLI(4,J),J=1,4),ELI(4)        SAT02000
    3 IF (IBCFNL.LT.0) WRITE (6,209)                                    SAT02010
      IF (IBCFNL.LT.0) GO TO 4                                          SAT02020
      WRITE (6,210)                                                     SAT02030
      WRITE (6,205) (OMEGL(1,J),J=1,4),(CAPLL(1,J),J=1,4),ELL(1)        SAT02040
      WRITE (6,206) (OMEGL(2,J),J=1,4),(CAPLL(2,J),J=1,4),ELL(2)        SAT02050
      WRITE (6,207) (OMEGL(3,J),J=1,4),(CAPLL(3,J),J=1,4),ELL(3)        SAT02060
      WRITE (6,208) (OMEGL(4,J),J=1,4),(CAPLL(4,J),J=1,4),ELL(4)        SAT02070
    4 WRITE (6,211)                                                     SAT02080
      IF ($DYNMC) GO TO 5                                               SAT02090
      WRITE (6,212) KMAX,MAXM,DELOAD,LSMAX,ITRMAX,LCHMAX,EPS            SAT02100
      WRITE (6,213) CHAR,TKN,ELAST,SIGC,NU                              SAT02110
      IF (.NOT.$PLOTS) GO TO 6                                          SAT02120
      ICHCK1=IABS(IRADII)+IABS(IGAMMA)+IABS(IOMEGS)+IABS(ICMEGT)        SAT02130
     1      +IABS(IDEOMS)+IABS(IBSTIF)+IABS(IDSTIF)+IABS(IBBSTF)        SAT02140
     2      +IABS(IDDSTF)+IABS(IPR)+IABS(IPS)+IABS(IPT)+IABS(ITT)       SAT02150
     3      +IABS(IMT)+IABS(IDTT)+IABS(IDMT)                            SAT02160
      IF (ICHCK1.GT.0) WRITE (6,214)                                    SAT02170
      IF (ICHCK1.EQ.0) GO TO 8                                          SAT02180
      IF (IRADII.NE.0) WRITE (6,215)                                    SAT02190
      IF (IGAMMA.NE.0) WRITE (6,216)                                    SAT02200
      IF (IOMEGS.NE.0) WRITE (6,217)                                    SAT02210
      IF (ICMEGT.NE.0) WRITE (6,218)                                    SAT02220
      IF (ICECMS.NE.0) WRITE (6,219)                                    SAT02230
      IF (IBSTIF.NE.0) WRITE (6,220)                                    SAT02240
      IF (IDSTIF.NE.0) WRITE (6,221)                                    SAT02250
      IF (IBBSTF.NE.0) WRITE (6,222)                                    SAT02260
      IF (IDDSTF.NE.0) WRITE (6,223)                                    SAT02270
      IF (IPR   .NE.0) WRITE (6,224)                                    SAT02280
```

```
      IF (IPS   .NE.0) WRITE (6,225)                                    SAT02290
      IF (IPT   .NE.0) WRITE (6,226)                                    SAT02300
      IF (ITT   .NE.0) WRITE (6,227)                                    SAT02310
      IF (IMT   .NE.0) WRITE (6,228)                                    SAT02320
      IF (ICTT  .NE.0) WRITE (6,229)                                    SAT02330
      IF (IDMT  .NE.0) WRITE (6,230)                                    SAT02340
    8 ICHCK1=IABS(INS)+IABS(INTH)+IABS(INSTH)+IABS(IQS)+IABS(IMS)       SAT02350
     1       +IABS(IMTH)+IABS(IMSTH)+IABS(IU)+IABS(IV)+IABS(IW)         SAT02360
     2       +IABS(IPHIS)+IABS(IPHIT)+IABS(IPHI)                        SAT02370
      IF((ICHCK1.GT.0).AND.$MCDAL) WRITE (6,231)                        SAT02380
      IF((ICHCK1.GT.0).AND..NOT.$MCDAL) WRITE (6,232)                   SAT02390
      IF (ICHCK1.EQ.0) GO TO 9                                          SAT02400
      IF (INS  .NE.0) WRITE (6,233)                                     SAT02410
      IF (INTH .NE.0) WRITE (6,234)                                     SAT02420
      IF (INSTH.NE.0) WRITE (6,235)                                     SAT02430
      IF (IQS  .NE.0) WRITE (6,236)                                     SAT02440
      IF (IMS  .NE.0) WRITE (6,237)                                     SAT02450
      IF (IMTH .NE.0) WRITE (6,238)                                     SAT02460
      IF (IMSTH.NE.0) WRITE (6,239)                                     SAT02470
      IF (IU   .NE.0) WRITE (6,240)                                     SAT02480
      IF (IV   .NE.0) WRITE (6,241)                                     SAT02490
      IF (IW   .NE.0) WRITE (6,242)                                     SAT02500
      IF (IPHIS.NE.0) WRITE (6,243)                                     SAT02510
      IF (IPHIT.NE.0) WRITE (6,244)                                     SAT02520
      IF (IPHI .NE.0) WRITE (6,245)                                     SAT02530
    9 GO TO 6                                                           SAT02540
    5 WRITE (6,246) KMAX,MAXM,DELOAD,LSMAX,ITRMAX,EPS                   SAT02550
      WRITE (6,247) CHAR,TKN,ELAST,SIGC,TEEO,NL                         SAT02560
    6 IF (NTHMAX.EQ.0) GO TO 7                                          SAT02570
      WRITE (6,248)                                                     SAT02580
      WRITE (6,249) (TH(NTH),NTH=1,NTHMAX)                              SAT02590
    7 IF (NDIMEN.EQ.1) WRITE (6,250)                                    SAT02600
      IF (NDIMEN.EQ.0) WRITE (6,251)                                    SAT02610
C***********************************************************************SAT02620
C     CALL APPROPRIATE CONTROLLING SUBPROGRAM                          *SAT02630
C***********************************************************************SAT02640
      IF ($DYNMC) CALL DYNAMC (P,CEE,CST,X,Z,ZC,Z2,Z3,ZDCT,IS,JS,ID,JD, SAT02650
     1PHIXB,PHITB)                                                      SAT02660
      IF (.NOT.$DYNMC) CALL STATIC (P,CEE,DST,X,Z,ZC,Z2,Z3,ZDCT,IS,JS,  SAT02670
     1ID,JD,PHIXB,PHITB)                                                SAT02680
C***********************************************************************SAT02690
C     READ STATEMENT FORMATS FOLLOW                                    *SAT02700
C***********************************************************************SAT02710
  100 FORMAT (18A4)                                                     SAT02720
  101 FORMAT (I5,L5,14I5)                                               SAT02730
  102 FORMAT (6E12.4)                                                   SAT02740
  103 FORMAT (2E12.4)                                                   SAT02750
  104 FORMAT (6E12.4)
```

```
  105 FCRMAT (4E16.8)                                                        SAT02770
  106 FCRMAT (2L2,29I2)                                                      SAT02780
  107 FCRMAT (I2)                                                            SAT02790
C************************************************************************SAT02800
C     WRITE STATEMENT FCRMATS FOLLOW                                        *SAT02810
C************************************************************************SAT02820
  200 FCRMAT ('1',48X,'--PRCBLEM NUMBER',I5,'--',///)                       SAT02830
  201 FCRMAT (' ',38X,18A4,///)                                             SAT02840
  202 FCRMAT (' ',49X,'--INPUT DATA RECCRD--',//'     THE BCUNDARY CCNDISAT02850
     1TICNS ARE: ',//)                                                      SAT02860
  203 FCRMAT (' ',T10,'THE SHELL HAS AN INITIAL PCLE',/)                    SAT02870
  204 FCRMAT (' ',T10,'AT THE INITIAL EDGE',//2X,'----------------CMEGA BSAT02880
     1AR----------------',15X,'----------------LAMDA BAR----------------',1SAT02890
     22X,'---EL-----',/)                                                    SAT02900
  205 FCRMAT (' ','(',4E10.3,')  NS          (',4E10.3,')  U         ',E10SAT02910
     1.3)                                                                   SAT02920
  206 FCRMAT (' ','(',4E10.3,')  NST   +     (',4E10.3,')  V    =    ',E10SAT02930
     1.3)                                                                   SAT02940
  207 FCRMAT (' ','(',4E10.3,')  CS          (',4E10.3,')  W         ',E10SAT02950
     1.3)                                                                   SAT02960
  208 FCRMAT (' ','(',4E10.3,')  PHIS        (',4E10.3,')  MS        ',E10SAT02970
     1.3)                                                                   SAT02980
  209 FCRMAT (' ',T10,'THE SHELL HAS A FINAL PCLE',/)                       SAT02990
  210 FCRMAT (' ',T10,'AT THE FINAL ECGE',//2X,'----------------OMEGA BARSAT03000
     1----------------',15X,'----------------LAMCA BAR----------------',12SAT03010
     2X,'---EL-----',/)                                                     SAT03020
  211 FCRMAT (////)                                                         SAT03030
  212 FCRMAT (' ',4X,'NUMBER OF STATIONS------------------------',I3/5X,'NSAT03040
     1UMBER OF MODES---------------------------',I3/5X,'INCREMENTAL LOAD FSAT03050
     2ACTCR-------------------',F6.3/5X,'MAXIMUM NUMBER CF LCAC STEPS-----SAT03060
     3-------',I3/5X,'MAXIMUM NUMBER OF ITERATICNS-------------',I3/5X,'MSAT03070
     4AXIMUM NUMBER OF LOAC FACTOR CHANGES---',I3/5X,'CCNVERGENCE CRITERSAT03080
     5ICN-------------------',F6.4//)                                       SAT03090
  213 FCRMAT (' ',4X,'CHARACTERISTIC SHELL DIMENSION-----------',E12.4/5XSAT03100
     1,'REFERENCE THICKNESS----------------------',E12.4/5X,'REFERENCE ELSAT03110
     2ASTICITY--------------------',E12.4/5X,'REFERENCE STRESS-----------SAT03120
     3---------------',E12.4/5X,'PCISSCNS RATIO--------------------------SAT03130
     4',E12.4//)                                                            SAT03140
  214 FCRMAT (' ',T10,'PLCTS HAVE BEEN REQUESTED FOR THE BELCW LISTED GESAT03150
     1CMETRY, STIFFNESS OR LOADING QUANTITIES:'/)                           SAT03160
  215 FCRMAT ('+',T70,'RADIUS'/)                                            SAT03170
  216 FCRMAT ('+',T70,'GAMMA'/)                                             SAT03180
  217 FCRMAT ('+',T70,'CMEGA-S'/)                                           SAT03190
  218 FCRMAT ('+',T70,'CMEGA-THETA'/)                                       SAT03200
  219 FCRMAT ('+',T70,'DECMEGA-S'/)                                         SAT03210
  220 FCRMAT ('+',T70,'B-STIFFNESS'/)                                       SAT03220
  221 FCRMAT ('+',T70,'C-STIFFNESS'/)                                       SAT03230
  222 FCRMAT ('+',T70,'B-PRIME'/)                                           SAT03240
```

```
  223 FORMAT ('+',T70,'D-PRIME'/)                                           SAT03250
  224 FORMAT ('+',T70,'PRESSURE LOADING - NORMAL'/)                         SAT03260
  225 FORMAT ('+',T70,'PRESSURE LOADING - MERIDIONAL'/)                     SAT03270
  226 FORMAT ('+',T70,'PRESSURE LOADING - CIRCUMFERENTIAL'/)                SAT03280
  227 FORMAT ('+',T70,'THERMAL LOADING'/)                                   SAT03290
  228 FORMAT ('+',T70,'THERMAL MOMENT'/)                                    SAT03300
  229 FORMAT ('+',T70,'D-THERMAL LOADING'/)                                 SAT03310
  230 FORMAT ('+',T70,'D-THERMAL MOMENT'/)                                  SAT03320
  231 FORMAT ('0',T20,'PLOTS HAVE BEEN REQUESTED FOR THE BELOW LISTED MOSAT03330
     1DAL QUANTITIES:'/)                                                    SAT03340
  232 FORMAT ('0',T20,'PLOTS HAVE BEEN REQUESTED FOR THE BELOW LISTED SUSAT03350
     1MMED QUANTITIES:'/)                                                   SAT03360
  233 FORMAT ('+',T70,'N-S'/)                                               SAT03370
  234 FORMAT ('+',T70,'N-THETA'/)                                           SAT03380
  235 FORMAT ('+',T70,'N-S-THETA'/)                                         SAT03390
  236 FORMAT ('+',T70,'Q-S'/)                                               SAT03400
  237 FORMAT ('+',T70,'M-S'/)                                               SAT03410
  238 FORMAT ('+',T70,'M-THETA'/)                                           SAT03420
  239 FORMAT ('+',T70,'M-S-THETA'/)                                         SAT03430
  240 FORMAT ('+',T70,'U'/)                                                 SAT03440
  242 FORMAT ('+',T70,'W'/)                                                 SAT03450
  241 FORMAT ('+',T70,'V'/)                                                 SAT03460
  243 FORMAT ('+',T70,'PHI-S'/)                                             SAT03470
  244 FORMAT ('+',T70,'PHI-THETA'/)                                         SAT03480
  245 FORMAT ('+',T70,'PHI'/)                                               SAT03490
  246 FORMAT (' ',4X,'NUMBER OF STATIONS----------------------',I3/5X,'NSAT03500
     1UMBER OF MODES--------------------------',I3/5X,'INCREMENTAL TIME--SAT03510
     2-----------------------',E9.4/5X,'MAXIMUM NUMBER OF TIME STEPS-----SAT03520
     3-------',I5/5X,'MAXIMUM NUMBER OF ITERATIONS------------',I3/5X,'CSAT03530
     4ONVERGENCE CRITERION--------------------',F6.4//)                     SAT03540
  247 FORMAT (' ',4X,'CHARACTERISTIC SHELL DIMENSION----------',E12.4/5XSAT03550
     1,'REFERENCE THICKNESS----------------------',E12.4/5X,'REFERENCE ELSAT03560
     2ASTICITY--------------------',E12.4/5X,'REFERENCE STRESS----------SAT03570
     3--------------',E12.4/5X,'REFERENCE TIME--------------------------SAT03580
     4',E12.4/5X,'POISSONS RATIO--------------------',E12.4//)              SAT03590
  248 FORMAT (' ',T20,'CIRCUMFERENTIAL COORDINATES FOR THE PRINT RECORD,SAT03600
     1 IN RADIAN MEASURE, ARE:'/)                                           SAT03610
  249 FORMAT (' ',T10,6F16.10)                                              SAT03620
  250 FORMAT (///' ',T20,'THE DATA PRINTED IS NON-DIMENSIONAL')             SAT03630
  251 FORMAT (///' ',T20,'THE DATA PRINTED IS DIMENSIONAL')                 SAT03640
      RETURN                                                                SAT03650
      END                                                                   SAT03660
      SUBROUTINE STATIC (P,DEE,DST,X,Z,ZC,Z2,Z3,ZDCT,IS,JS,ID,JD,PHIXB,     SAT03670
     1PHITB)                                                                SAT03680
C***********************************************************************SAT03690
C     THIS SUBROUTINE IS THE MAJOR CONTROLLING ROUTINE FOR ALL STATIC  *SAT03700
C     ANALYSIS PROBLEMS.  IT CONDUCTS INITIALIZATION, PREPARES INPUT   *SAT03710
C     MATERIAL FOR PLOTTING IF REQ'D, DETERMINES SHELL GEOMETRY, STIFF-*SAT03720
```

```
C      NESSES,LOADING (PHYSICAL AND/OR THERMAL), AND INITIAL CONDITIONS.*SAT03730
C      IT CONTROLS PROBLEM SOLUTION PROCEDURE.                          *SAT03740
C*****************************************************************************SAT03750
       IMPLICIT LOGICAL*1 ($)                                            SAT03760
       REAL*4 NU,LAM,LAM2,JAY,MT,LSD18,LSD1N,MASS                        SAT03770
       DIMENSION P(4,4,1),DEE(4,4,1),DST(4,4,1),X(4,1),Z(4,1),           SAT03780
      1ZC(4,1),Z2(4,1),Z3(4,1),ZDOT(4,1),IS(99,1),JS(99,1),IC(99,1),     SAT03790
      2JC(99,1),PHIXB(1),PHITB(1)                                        SAT03800
       COMMON /IBL1/ MNMAX                                               SAT03810
       COMMON /IBL2/ N(99),MNINIT                                        SAT03820
C                    'N' APPEARS AS 'NN' IN SUBROUTINES PLOAD & EFG      SAT03830
       COMMON /IBL3/ MO,M1,M2,M3                                         SAT03840
       COMMON /IBL4/ KMAX,KL                                             SAT03850
       COMMON /IBL5/ IBCINL,IBCFNL                                       SAT03860
       COMMON /IBL6/ KLL                                                 SAT03870
       COMMON /IBL7/ MNMAXC,MAXD(99),MAXS(99),MAXSY(99),IJS(99)          SAT03880
       COMMON /IBL8/ LSTEP,ITR                                           SAT03890
       COMMON /IBL9/ MAXM                                                SAT03900
       COMMON /IBL10/ IFREQ,NTHMAX                                       SAT03910
       COMMON /IBL11/ ICORFL,IPASS                                       SAT03920
       COMMON /IBL12/ KMAX1,KMAX2,NCONV                                  SAT03930
       COMMON /IBL13/ ITRMAX,LSMAX                                       SAT03940
       COMMON /BL1/  A(4,4),BEE(4,4),C(4,4)                              SAT03950
       COMMON /BL3/  PR(99),PX(99),PT(99)                                SAT03960
       COMMON /BL4/  ZF1M(4,4,99),ZF2M(4,4,99),
      1              ZF3M(4,4,99),ZF4M(4,4,99)
       COMMON /BL5/  TT(99),MT(99),DT(99),DMT(99)                        SAT03990
C                    'MT' APPEARS AS 'EMT' IN SUBROUTINES INLPOL & FNLPOLSAT04000
       COMMON /BL6/  SOE,CSE,ALOAD                                       SAT04010
       COMMON /BL7/  D1,S1                                               SAT04020
       COMMON /BL8/  R(500),GAM(500),OMT(500)                            SAT04030
       COMMON /BL9/  FFS(4,99),EL1S(4),GEES(4,99)
       COMMON /BL10/ PHIX(99),PHIT(99),PHI(99)                           SAT04050
       COMMON /BL11/ OMXI(500),PHEE,TO,T2                                SAT04060
       COMMON /BL12/ TDLI,TDEL                                           SAT04070
       COMMON /BL13/ OMEGI(4,4),CAPL1(4,4),OMEGL(4,4),CAPLL(4,4),        SAT04080
      1              UNIT(4,4)                                           SAT04090
       COMMON /BL14/ LAM2,LSD18,LSD1N                                    SAT04100
       COMMON /BL15/ NU,U1(99),V1(99),W1(99),V2(99),U2(99),W2(99),U3(99),SAT04110
      1              V3(99),W3(99)                                       SAT04120
       COMMON /BL16/ EPS                                                 SAT04130
       COMMON /BL17/ DEL                                                 SAT04140
       COMMON /BL18/ EL1(4),ELL(4)                                       SAT04150
       COMMON /BL19/ TH(36)                                              SAT04160
       COMMON /BL20/ DEOMX(500)                                          SAT04170
       COMMON /BL23/ JAY(4,4),H(4,4)                                     SAT04180
       COMMON /BL24/ DL(4,4,99),DG(4,4,99),DF(4,4,99)
       COMMON /BL25/ E(4,4),F(4,4),G(4,4)                                SAT04200
```

```
      COMMON /BL27/ BX3(99),BT3(99),BXT3(99),BE3(99)                    SAT04210
      COMMON /BL28/ EXX3(99),ETT3(99),ETX3(99),EXT3(99),EX3(99),ET3(99) SAT04220
      COMMON /BL29/ BX1(99),BT1(99),BXT1(99),BE1(99),BX2(99),BT2(99),   SAT04230
     1              BXT2(99),BE2(99)                                    SAT04240
      COMMON /BL30/ EXX1(99),ETT1(99),ETX1(99),EX1(99),ET1(99),EXX2(99),SAT04250
     1              ETT2(99),ETX2(99),EXT2(99),EX2(99),ET2(99)          SAT04260
      COMMON /BL31/ DELSQ,EXT1(99)                                      SAT04270
      COMMON /BL32/ TKN,ELAST,CHAR,SIGC                                 SAT04280
      COMMON /BL100/ TEEO,$DYNMC                                        SAT04290
      COMMON /BL101/ DELSD                                              SAT04300
      COMMON /BL102/ DELCAD                                             SAT04310
      COMMON /BL103/ MASS(500)                                          SAT04320
      COMMON /BL110/ TX(99),TTH(99),TXT(99),MX(99),MTH(99),MXT(99),     SAT04330
     1              QS(99)                                              SAT04340
      COMMON /BL111/ ABZ,ABZO,ABZN,ABZ3,CD2                             SAT04350
      COMMON /BLPHS/ PHX(99),PHT(99)
      COMMON /BLPLOT/  IRADII,IGAMMA,IOMEGS,IOMEGT,IDECMS,IBSTIF,IDSTIF, SAT04360
     1                 IBBSTF,IDDSTF,IPR,IPS,IPT,ITT,IMT,IDTT,IDMT,INS,  SAT04370
     2                 INTH,INSTH,IQS,IMS,IMTH,IMSTH,IU,IV,IW,IPHIS,     SAT04380
     3                 IPHIT,IPHI,$PLOTS,$MODAL                          SAT04390
      COMMON /BLPLT1/  XRADII(200),YGAMMA(200),YOMEGS(200),YOMEGT(200),  SAT04400
     1                 YDECMS(200),YBSTIF(200),YDSTIF(200),YBBSTF(200),  SAT04410
     2                 YDDSTF(200),YPR(200),YPS(200),YPT(200),YTT(200),  SAT04420
     3                 YMT(200),YDTT(200),YDMT(200),YNS(200),YNTH(200),  SAT04430
     4                 YNSTH(200),YQS(200),YMS(200),YMTH(200),YMSTH(200),SAT04440
     5                 YU(200),YV(200),YW(200),YPHIS(200),YPHIT(200),    SAT04450
     6                 YPHI(200),XSTATN(200)                             SAT04460
      COMMON /BLDATA/  TITLE,NO,IMODE,NDIMEN,IPRINT,LCHMAX,IC           SAT04470
      DIMENSION SIGT(2),SIGC(2),TITLE(18)                               SAT04480
C***********************************************************************SAT04490
      WRITE (6,8888)                                                    SAT04500
      DELSD=DELCAD*DELOAD                                               SAT04510
      KL=KMAX-1                                                         SAT04520
      KLL=KMAX-2                                                        SAT04530
      KMAX1=KMAX+1                                                      SAT04540
      KMAX2=KMAX+2                                                      SAT04550
      AK=KL                                                             SAT04560
      SIGT(1)=SIGO*TKN                                                  SAT04570
      SIGT(2)=SIGO/ELAST                                                SAT04580
      SIGC(1)=SIGO*CHAR/ELAST                                           SAT04590
      SIGC(2)=SIGO*TKN**3/CHAR                                          SAT04600
      IF (IBCINL.LT.0) GO TO 14                                         SAT04610
      DO 98 I=1,4                                                       SAT04620
      DO 98 J=1,4                                                       SAT04630
      KKLM=J/4+1                                                        SAT04640
      OMEG1(I,J)=OMEG1(I,J)*SIGT(KKLM)                                  SAT04650
   98 CAPL1(I,J)=CAPL1(I,J)*SIGC(KKLM)                                  SAT04660
   14 IF(IBCFNL.LT.0) GO TO 17                                          SAT04670
```

```
      DO 99 I=1,4                                                       SAT04680
      DO 99 J=1,4                                                       SAT04690
      KKLM=J/4+1                                                        SAT04700
      OMEGL(I,J)=OMEGL(I,J)*SIGT(KKLM)                                  SAT04710
   99 CAPLL(I,J)=CAPLL(I,J)*SIGC(KKLM)                                  SAT04720
   17 LAM=TKN/CHAR                                                      SAT04730
      SOE=SIGC/ELAST                                                    SAT04740
      OSE=.5*SOE                                                        SAT04750
      D1=1.0-NU                                                         SAT04760
      S1=1.0+NU                                                         SAT04770
      LAM2=LAM**2                                                       SAT04780
      IF(NDIMEN.LT.1) GO TO 228                                         SAT04790
      SIGC=1.0                                                          SAT04800
      ELAST=1.0                                                         SAT04810
      TKN=1.0                                                           SAT04820
      CHAR=1.0                                                          SAT04830
  228 DO 230 M=1,MAXM                                                   SAT04840
      PHX(M)=0.0
      PHT(M)=0.0
      N(M)=0.0                                                          SAT04850
      PX(M)=0.0                                                         SAT04860
      PT(M)=0.0                                                         SAT04870
      PR(M)=0.0                                                         SAT04880
      TT(M)=0.0                                                         SAT04890
      MT(M)=0.0                                                         SAT04900
      DT(M)=0.0                                                         SAT04910
      DMT(M)=0.0                                                        SAT04920
      MAXD(M)=0                                                         SAT04930
      MAXS(M)=0                                                         SAT04940
  230 MAXSY(M)=0                                                        SAT04950
      CALL GEOM                                                         SAT04960
      ICHCK1=IABS(IGAMMA)+IABS(IOMEGS)+IABS(IOMEGT)+IABS(IDEOMS)        SAT04970
     1       +IABS(IRADII)                                              SAT04980
      ICHCK2=IABS(IBSTIF)+IABS(IDSTIF)+IABS(IBBSTF)+IABS(IDDSTF)        SAT04990
      IF (.NOT.$PLOTS) GO TO 1001                                       SAT05000
      DO 2 K=1,KMAX                                                     SAT05010
    2 XSTATN(K)=FLOAT(K)                                                SAT05020
      IF (ICHCK1.EQ.0) GO TO 1001                                       SAT05030
      DO 1 K=1,KMAX                                                     SAT05040
      XRADII(K)=R(K)*CHAR                                               SAT05050
      YGAMMA(K)=GAM(K)/CHAR                                             SAT05060
      YOMEGS(K)=OMXI(K)/CHAR                                            SAT05070
      YOMEGT(K)=OMT(K)/CHAR                                             SAT05080
      YDEOMS(K)=DEOMX(K)/(CHAR*CHAR)                                    SAT05090
    1 CONTINUE                                                          SAT05100
 1001 CONTINUE                                                          SAT05110
      DO 86 K=1,KMAX                                                    SAT05120
   86 MASS(K)=0.                                                        SAT05130
```

```
      WRITE(6,802)                                                      SAT05140
      DO 978 K=1,KMAX                                                   SAT05150
      RKK=R(K)*CHAR                                                     SAT05160
      CMXIK=CMXI(K)/CHAR                                                SAT05170
      GAMK=GAM(K)/CHAR                                                  SAT05180
      CMTK=CMT(K)/CHAR                                                  SAT05190
      DEOMXK=DEOMX(K)/(CHAR*CHAR)                                       SAT05200
  978 WRITE(6,803) K,RKK,GAMK,CMXIK,CMTK,DEOMXK                         SAT05210
  805 MC=0                                                              SAT05220
      M1=0                                                              SAT05230
      M2=0                                                              SAT05240
      M3=0                                                              SAT05250
      ABN=CHAR/SIGO/TKN                                                 SAT05260
      ZN=SIGO*TKN                                                       SAT05270
      WRITE(6,112)                                                      SAT05280
      DO 888 K=1,KMAX                                                   SAT05290
      CALL BDE(K,B,DB,D,DD)                                             SAT05300
      BST=ELAST*TKN                                                     SAT05310
      ZST=ELAST*TKN**3                                                  SAT05320
      B=B*BST                                                           SAT05330
      D=D*ZST                                                           SAT05340
      DB=DB/CHAR*BST                                                    SAT05350
      DD=DD/CHAR*ZST                                                    SAT05360
      WRITE (6,71) K,B,D,DB,DD                                          SAT05370
      IF (.NOT.$PLOTS.OR.(ICHCK2.EQ.0)) GO TO 888                       SAT05380
      YBSTIF(K)=B                                                       SAT05390
      YDSTIF(K)=D                                                       SAT05400
      YBBSTF(K)=DB                                                      SAT05410
      YDDSTF(K)=DD                                                      SAT05420
  888 CONTINUE                                                          SAT05430
      CALL PLOAD(1,Z)                                                   SAT05440
      CALL TLOAD(1,Z)                                                   SAT05450
      DO 889 M=1,MNMAX                                                  SAT05460
      WRITE(6,113) N(M)                                                 SAT05470
      WRITE(6,114)                                                      SAT05480
      ICHCK3=IABS(IPR)+IABS(IPS)+IABS(IPT)+IABS(ITT)+IABS(IMT)          SAT05490
     1        +IABS(IDTT)+IABS(IDMT)                                    SAT05500
      DO 890 K=1,KMAX                                                   SAT05510
      CALL PLOAD(K,Z)                                                   SAT05520
      CALL TLOAD(K,Z)                                                   SAT05530
      PRM=PR(M)/ABN                                                     SAT05540
      PTM=PT(M)/ABN                                                     SAT05550
      PXM=PX(M)/ABN                                                     SAT05560
      TTM=TT(M)*ZN                                                      SAT05570
      EMTM=MT(M)/CHAR*ZN*TKN*TKN                                        SAT05580
      DTM=DT(M)/CHAR*ZN                                                 SAT05590
      DMTM=DMT(M)*ZN*TKN*TKN/(CHAR*CHAR)                                SAT05600
      WRITE (6,115) K,PRM,PXM,PTM,TTM,EMTM,DTM,DMTM                     SAT05610
```

```
      IF (.NOT.$PLOTS.OR.(ICHCK3.EQ.0)) GO TO 890                      SAT05620
      YPR(K)=PRM                                                       SAT05630
      YPS(K)=PXM                                                       SAT05640
      YPT(K)=PTM                                                       SAT05650
      YTT(K)=TTM                                                       SAT05660
      YMT(K)=EMTM                                                      SAT05670
      YDTT(K)=DTM                                                      SAT05680
      YDMT(K)=DMTM                                                     SAT05690
  890 CONTINUE                                                         SAT05700
      IF (M.EQ.1) ICHCK3=ICHCK1+ICHCK2+ICHCK3                          SAT05710
      IF ($PLOTS.AND.(ICHCK3.GT.0)) CALL PLOT1(M)                      SAT05720
  889 CONTINUE                                                         SAT05730
      DELSQ=DEL**2                                                     SAT05740
      TDLI=.5/DEL                                                      SAT05750
      TDEL=2.0*DEL                                                     SAT05760
      MNINIT=1                                                         SAT05770
      MNMAXC=MNMAX                                                     SAT05780
      DO 20 I=1,4                                                      SAT05790
      DO 20 J=1,4                                                      SAT05800
      UNIT(I,J)=0.0                                                    SAT05810
   20 IF(I.EQ.J) UNIT(I,J)=1.0                                         SAT05820
      NMAX=MAXM*KMAX2                                                  SAT05830
      DO 22 K=1,NMAX                                                   SAT05840
      DO 22 I=1,4                                                      SAT05850
      ZDOT(I,K)=0.0                                                    SAT05860
      ZC(I,K)=0.0                                                      SAT05870
      Z2(I,K)=0.0                                                      SAT05880
      Z3(I,K)=0.0                                                      SAT05890
   22 Z(I,K)=0.0                                                       SAT05900
      ALOAD=DELOAD                                                     SAT05910
      CALL IMPERF (PHIXB,PHITB)                                        SAT05920
      CALL PMATRX (P,X,ZC,Z2,Z3,DEE,DST)                               SAT05930
      LSTEP=1                                                          SAT05940
      LCHANG=0                                                         SAT05950
      ITR=1                                                            SAT05960
      ICORFL=0                                                         SAT05970
      IF(MNMAX.EQ.MAXM) ICORFL=1                                       SAT05980
      IPASS=0                                                          SAT05990
  400 CALL XANDZ (P,DEE,DST,X,Z,ZC,Z2,Z3,ZDOT, IS,JS,ID,JD,PHIXB,PHITB) SAT06000
      IF(ITRMAX.EQ.1) GO TO 50                                         SAT06010
      MNMAXC=MNMAX                                                     SAT06020
      IF(IPASS.LT.2) CALL MODES (IS,JS,ID,JD,P,X,ZC,Z2,Z3,DEE,DST)
      IF((NCONV.EQ.1).AND.(ITR.GT.1)) GO TO 50                         SAT06040
      IF(ITR.LT.ITRMAX) GO TO 23                                       SAT06050
      IF(LCHANG.LT.LCHMAX) GO TO 30                                    SAT06060
      WRITE(6,220) NO                                                  SAT06070
      GO TO 500                                                        SAT06080
   50 FL=LSTEP                                                         SAT06090
```

```
      FI=IPRINT                                                         SAT06100
      LI=LSTEP/IPRINT                                                   SAT06110
      FLI=LI                                                            SAT06120
      FT=FLI-FL/FI                                                      SAT06130
      IF(FT.EQ.0.) CALL  OUTPUT(IMODE,P,DEE,DST,X,Z,ZO,Z2,Z3,ZDOT,IS,JS, SAT06140
     1ID,JD,PHIXB,PHITB)                                                SAT06150
      IF(LSTEP.EQ.1) ITR=1                                              SAT06160
      IF(LSTEP.EQ.1) ITRPR=1                                            SAT06170
      IF(ITR.GT.ITRPR)  ITRPR=ITR                                       SAT06180
      IF(LSTEP.GE.LSMAX) GO TO 360                                      SAT06190
   60 DO 61 MN=1,MNMAXO                                                 SAT06200
      DO 61 K=1,KMAX2                                                   SAT06210
      IK=K+(MN-1)*KMAX2                                                 SAT06220
      DO 61 I=1,4                                                       SAT06230
      ZN=2.0*Z(I,IK)-ZO(I,IK)                                           SAT06240
      ZO(I,IK)=Z(I,IK)                                                  SAT06250
   61 Z(I,IK)=ZN                                                        SAT06260
      IF(LSTEP.GE.LSMAX) GO TO 360                                      SAT06270
   62 ALOAD=ALOAD+DELOAD                                                SAT06280
      LSTEP=LSTEP+1                                                     SAT06290
      ITR=1                                                             SAT06300
      GO TO 400                                                         SAT06310
  360 WRITE(6,221) NO                                                   SAT06320
      GO TO 500                                                         SAT06330
   23 ITR=ITR+1                                                         SAT06340
      GO TO 400                                                         SAT06350
   30 IF(LSTEP-1) 310,310,320                                           SAT06360
  310 WRITE(6,223)                                                      SAT06370
      GO TO 500                                                         SAT06380
  320 WRITE(6,222)                                                      SAT06390
      LCHANG=LCHANG+1                                                   SAT06400
      LSTEP=LSTEP-1                                                     SAT06410
      ALOAD=ALOAD-DELOAD                                                SAT06420
      DELOAD=DELOAD/5.0                                                 SAT06430
      DO 32 MN=1,MNMAXO                                                 SAT06440
      DO 32 K=1,KMAX2                                                   SAT06450
      IK=K+(MN-1)*KMAX2                                                 SAT06460
      DO 32 I=1,4                                                       SAT06470
   32 Z(I,IK)=ZO(I,IK)                                                  SAT06480
      GO TO 62                                                          SAT06490
C***********************************************************************SAT06500
   71 FORMAT(20X,I3,4X,4E20.6)                                          SAT06510
  112 FORMAT(////17X,12H   STATION    20H        B STIFFNESS    20H        D STSAT06520
     1IFFNESS    20H         B PRIME          20H         D PRIME           //)  SAT06530
  113 FORMAT(///25X,44HPRESSURE AND TEMPERATURE COEFFICIENTS FOR N=I3,8HSAT06540
     1  FOLLOW//)                                                       SAT06550
  114 FORMAT(5X,7HSTATION,3X,15H          PR        15H        PX        15H    SAT06560
     1  PT          15H          TT          15H        MT          15H        DTT          1SAT06570
```

```
    25H       DMT       //)                                              SAT06580
115 FORMAT(6X,I3,7X,7E15.4)                                              SAT06590
220 FORMAT(1H1,80H            THE MAXIMUM NUMBER OF LOAD CHANGES HAVE BESAT06600
   1EN MADE.  END PROBLEM NUMBERI4)                                      SAT06610
221 FORMAT(1H1,79H            THE MAXIMUM NUMBER OF LOAD STEPS HAVE BEENSAT06620
   1 TAKEN.  END PROBLEM NUMBERI4)                                       SAT06630
222 FORMAT(1H1,119H           THE SOLUTION DID NOT CONVERGE WITHIN THE MSAT06640
   1AXIMUM NUMBER OF ITERATIONS.  THE LOAD FACTOR HAS BEEN DIVIDED BY   SAT06650
   25.)                                                                  SAT06660
223 FORMAT(1H1,69H            THE SOLUTION DID NOT CONVERGE FOR THE FIRSSAT06670
   1T LOAD INCREMENT./11X,71HLOOK FOR AN ERROR IN THE INPUT DATA, OR TSAT06680
   2RY A SMALLER VALUE FOR DELOAD.)                                      SAT06690
802 FORMAT(1H1,17X,15H STATION          16H      RADIUS      16H     GAMMA SAT06700
   1      16H      OMEGA S     16H      OMEGA THETA16H     DOMEGA S  //) SAT06710
803 FORMAT(20X,I3, 9X,5E16.4)                                            SAT06720
8888 FORMAT ('0',T20,'EXECUTING IN SUBROUTINE "STATIC"')                 SAT06730
500 RETURN                                                               SAT06740
    END                                                                  SAT06750
    SUBROUTINE DYNAMC (P,CEE,DST,X,Z,ZD,Z2,Z3,ZDOT,IS,JS,ID,JD,PHIXB,    SAT06760
   1PHITB)                                                               SAT06770
C***********************************************************************SAT06780
C     THIS SUBROUTINE IS ONE OF THE MAJOR CONTROLLING SUBROUTINES FOR  *SAT06790
C     ALL DYNAMIC ANALYSIS PROBLEMS.  IT OPERATES IN A FASHION SIMILAR *SAT06800
C     TO SUBROUTINE STATIC.                                            *SAT06810
C***********************************************************************SAT06820
    IMPLICIT LOGICAL*1 ($)                                               SAT06830
    REAL*4 NU,LAM,LAM2,JAY,MT,LSD18,LSDIN,MASS                           SAT06840
    DIMENSION P(4,4,1),CEE(4,4,1),DST(4,4,1),X(4,1),Z(4,1),              SAT06850
   1ZD(4,1),Z2(4,1),Z3(4,1),ZDOT(4,1),IS(99,1),JS(99,1),ID(99,1),        SAT06860
   2JD(99,1),PHIXB(1),PHITB(1)                                           SAT06870
    COMMON /IBL1/ MNMAX                                                  SAT06880
    COMMON /IBL2/ N(99),MNINIT                                           SAT06890
C               'N' APPEARS AS 'NN' IN SUBROUTINES PLOAD & EFG           SAT06900
    COMMON /IBL3/ M0,M1,M2,M3                                            SAT06910
    COMMON /IBL4/ KMAX,KL                                                SAT06920
    COMMON /IBL5/ IBCINL,IBCFNL                                          SAT06930
    COMMON /IBL6/ KLL                                                    SAT06940
    COMMON /IBL7/ MNMAXO,MAXD(99),MAXS(99),MAXSY(99),IJS(99)             SAT06950
    COMMON /IBL8/ LSTEP,ITR                                              SAT06960
    COMMON /IBL9/ MAXM                                                   SAT06970
    COMMON /IBL10/ IFREQ,NTHMAX                                          SAT06980
    COMMON /IBL11/ ICORFL,IPASS                                          SAT06990
    COMMON /IBL12/ KMAX1,KMAX2,NCONV                                     SAT07000
    COMMON /IBL13/ ITRMAX,LSMAX                                          SAT07010
    COMMON /BL1/  A(4,4),BEE(4,4),C(4,4)                                 SAT07020
    COMMON /BL3/  PR(99),PX(99),PT(99)                                   SAT07030
    COMMON /BL4/  ZF1M(4,4,99),ZF2M(4,4,99),
   1              ZF3M(4,4,99),ZF4M(4,4,99)
```

```
      COMMON /BL5/  TT(99),MT(99),DT(99),DMT(99)                        SAT07060
C                   'MT' APPEARS AS 'EMT' IN SUBROUTINES INLPCL & FNLPOLSAT07070
      COMMON /BL6/  SOE,CSE,ALOAD                                       SAT07080
      COMMON /BL7/  D1,S1                                               SAT07090
      COMMON /BL8/  R(500),GAM(500),OMT(500)                            SAT07100
      COMMON /BL9/  FFS(4,99),ELIS(4),GEES(4,99)
      COMMON /BL10/ PHIX(99),PHIT(99),PHI(99)                           SAT07120
      COMMON /BL11/ OMXI(500),PHEE,TO,T2                                SAT07130
      COMMON /BL12/ TDLI,TDEL                                           SAT07140
      COMMON /BL13/ OMEG1(4,4),CAPL1(4,4),OMEGL(4,4),CAPLL(4,4),        SAT07150
     1              UNIT(4,4)                                           SAT07160
      COMMON /BL14/ LAM2,LSD18,LSD1N                                    SAT07170
      COMMON /BL15/ NU,U1(99),V1(99),W1(99),V2(99),L2(99),W2(99),U3(99),SAT07180
     1              V3(99),W3(99)                                       SAT07190
      COMMON /BL16/ EPS                                                 SAT07200
      COMMON /BL17/ DEL                                                 SAT07210
      COMMON /BL18/ EL1(4),ELL(4)                                       SAT07220
      COMMON /BL19/ TH(36)                                              SAT07230
      COMMON /BL20/ DEOMX(500)                                          SAT07240
      COMMON /BL23/ JAY(4,4),H(4,4)                                     SAT07250
      COMMON /BL24/ DL(4,4,99),DG(4,4,99),DF(4,4,99)
      COMMON /BL25/ E(4,4),F(4,4),G(4,4)                                SAT07270
      COMMON /BL27/ BX3(99),BT3(99),BXT3(99),BE3(99)                    SAT07280
      COMMON /BL28/ EXX3(99),ETT3(99),ETX3(99),EXT3(99),EX3(99),ET3(99) SAT07290
      COMMON /BL29/ BX1(99),BT1(99),BXT1(99),BE1(99),BX2(99),BT2(99),   SAT07300
     1              BXT2(99),BE2(99)                                    SAT07310
      COMMON /BL30/ EXX1(99),ETT1(99),ETX1(99),EX1(99),ET1(99),EXX2(99),SAT07320
     1              ETT2(99),ETX2(99),EXT2(99),EX2(99),ET2(99)          SAT07330
      COMMON /BL31/ DELSQ,EXT1(99)                                      SAT07340
      COMMON /BL32/ TKN,ELAST,CHAR,SIGC                                 SAT07350
      COMMON /BL100/ TEEC,$DYNMC                                        SAT07360
      COMMON /BL101/ DELSD                                              SAT07370
      COMMON /BL102/ DELOAD                                             SAT07380
      COMMON /BL103/ MASS(500)                                          SAT07390
      COMMON /BL110/ TX(99),TTH(99),TXT(99),MX(99),MTH(99),MXT(99),     SAT07400
     1              QS(99)                                              SAT07410
      COMMON /BL111/ ABZ,ABZO,ABZN,ABZ3,CC2                             SAT07420
      COMMON /BLPHS/ PHX(99),PHT(99)
      COMMON /BLFLOT/  IRADII,IGAMMA,IOMEGS,IOMEGT,IDEOMS,IBSTIF,IDSTIF, SAT07430
     1                 IBBSTF,IDDSTF,IPR,IPS,IPT,ITT,IMT,IDTT,IDMT,INS, SAT07440
     2                 INTH,INSTH,IQS,IMS,IMTH,IMSTH,IU,IV,IW,IPHIS,    SAT07450
     3                 IPHIT,IPHI,$PLOTS,$MODAL                         SAT07460
      COMMON /BLPLT1/  XRADII(200),YGAMMA(200),YOMEGS(200),YOMEGT(200), SAT07470
     1                 YDEOMS(200),YBSTIF(200),YDSTIF(200),YBBSTF(200), SAT07480
     2                 YDDSTF(200),YPR(200),YPS(200),YPT(200),YTT(200), SAT07490
     3                 YMT(200),YDTT(200),YDMT(200),YNS(200),YNTH(200), SAT07500
     4                 YNSTH(200),YQS(200),YMS(200),YMTH(200),YMSTH(200),SAT07510
     5                 YU(200),YV(200),YW(200),YPHIS(200),YPHIT(200),   SAT07520
```

```
     6              YPHI(200),XSTATN(200)                                SAT07530
      COMMON /BLDATA/ TITLE,NC,IMODE,NDIMEN,IPRINT,LCHMAX,IC             SAT07540
      DIMENSION SIGT(2),SIGC(2),TITLE(18)                                SAT07550
      DIMENSION VBAR(99),AVB(99)                                         SAT07560
C************************************************************************SAT07570
      WRITE (6,8888)                                                     SAT07580
      DELSD=DELOAD*DELCAD                                                SAT07590
      KL=KMAX-1                                                          SAT07600
      KLL=KMAX-2                                                         SAT07610
      KMAX1=KMAX+1                                                       SAT07620
      KMAX2=KMAX+2                                                       SAT07630
      AK=KL                                                              SAT07640
      SIGT(1)=SIGO*TKN                                                   SAT07650
      SIGT(2)=SIGO/ELAST                                                 SAT07660
      SIGC(1)=SIGO*CHAR/ELAST                                            SAT07670
      SIGC(2)=SIGO*TKN**3/CHAR                                           SAT07680
      IF (IBCINL.LT.0) GO TO 14                                          SAT07690
      DO 98 I=1,4                                                        SAT07700
      DO 98 J=1,4                                                        SAT07710
      KKLM=J/4+1                                                         SAT07720
      OMEG1(I,J)=OMEG1(I,J)*SIGT(KKLM)                                   SAT07730
   98 CAPL1(I,J)=CAPL1(I,J)*SIGC(KKLM)                                   SAT07740
   14 IF(IBCFNL.LT.0) GO TO 17                                           SAT07750
      DO 99 I=1,4                                                        SAT07760
      DO 99 J=1,4                                                        SAT07770
      KKLM=J/4+1                                                         SAT07780
      OMEGL(I,J)=OMEGL(I,J)*SIGT(KKLM)                                   SAT07790
   99 CAPLL(I,J)=CAPLL(I,J)*SIGC(KKLM)                                   SAT07800
   17 LAM=TKN/CHAR                                                       SAT07810
      SCE=SIGO/ELAST                                                     SAT07820
      CSE=.5*SCE                                                         SAT07830
      D1=1.0-NU                                                          SAT07840
      S1=1.0+NU                                                          SAT07850
      LAM2=LAM**2                                                        SAT07860
      IF(NDIMEN.LT.1) GO TO 228                                          SAT07870
      SIGO=1.0                                                           SAT07880
      ELAST=1.0                                                          SAT07890
      TKN=1.0                                                            SAT07900
      CHAR=1.0                                                           SAT07910
  228 DO 230 M=1,MAXM                                                    SAT07920
      PHX(M)=0.0
      PHT(M)=0.0
      N(M)=0.0                                                           SAT07930
      PX(M)=0.0                                                          SAT07940
      PT(M)=0.0                                                          SAT07950
      PR(M)=0.0                                                          SAT07960
      TT(M)=0.0                                                          SAT07970
      MT(M)=0.0                                                          SAT07980
```

```
      DT(M)=0.0
      DMT(M)=0.0
      MAXD(M)=0
      MAXS(M)=0
  230 MAXSY(M)=0
      CALL GEOM
      ICHCK1=IABS(IGAMMA)+IABS(IOMEGS)+IABS(IOMEGT)+IABS(IDEOMS)
     1      +IABS(IRADII)
      ICHCK2=IABS(IBSTIF)+IABS(IDSTIF)+IABS(IBBSTF)+IABS(IDDSTF)
      IF (.NOT.$PLOTS) GO TO 1001
      DO 2 K=1,KMAX
    2 XSTATN(K)=FLOAT(K)
      IF (ICHCK1.EQ.0) GO TO 1001
      DO 1 K=1,KMAX
      XRADII(K)=R(K)*CHAR
      YGAMMA(K)=GAM(K)/CHAR
      YOMEGS(K)=OMXI(K)/CHAR
      YOMEGT(K)=OMT(K)/CHAR
      YDEOMS(K)=DEOMX(K)/(CHAR*CHAR)
    1 CONTINUE
 1001 CONTINUE
  804 WRITE(6,810)
      TEED=TEEO
      IF(NDIMEN.EQ.1) TEED=1.0
      DO 979 K=1,KMAX
      RKK=R(K)*CHAR
      OMXIK=OMXI(K)/CHAR
      GAMK=GAM(K)/CHAR
      OMTK=OMT(K)/CHAR
      DEOMXK=DEOMX(K)/(CHAR*CHAR)
      AMSS=MASS(K)*TEED**2*ELAST*TKN/CHAR**2
  979 WRITE(6,813) K,RKK,GAMK,OMXIK,OMTK,DEOMXK,AMSS
  805 MC=0
      M1=0
      M2=0
      M3=0
      AEN=CHAR/SIGO/TKN
      ZN=SIGO*TKN
      WRITE(6,112)
      DO 888 K=1,KMAX
      CALL BDB(K,B,DB,D,DD)
      BST=ELAST*TKN
      ZST=ELAST*TKN**3
      B=B*BST
      D=D*ZST
      DB=DB/CHAR*BST
      DD=DD/CHAR*ZST
      WRITE (6,71) K,B,D,DB,DD
```

SAT07990
SAT08000
SAT08010
SAT08020
SAT08030
SAT08040
SAT08050
SAT08060
SAT08070
SAT08080
SAT08090
SAT08100
SAT08110
SAT08120
SAT08130
SAT08140
SAT08150
SAT08160
SAT08170
SAT08180
SAT08190
SAT08200
SAT08210
SAT08220
SAT08230
SAT08240
SAT08250
SAT08260
SAT08270
SAT08280
SAT08290
SAT08300
SAT08310
SAT08320
SAT08330
SAT08340
SAT08350
SAT08360
SAT08370
SAT08380
SAT08390
SAT08400
SAT08410
SAT08420
SAT08430
SAT08440
SAT08450
SAT08460

```
      IF (.NOT.$PLOTS.OR.(ICHCK2.EQ.0)) GO TO 888                       SAT08470
      YESTIF(K)=B                                                       SAT08480
      YDSTIF(K)=D                                                       SAT08490
      YBBSTF(K)=DB                                                      SAT08500
      YDDSTF(K)=DD                                                      SAT08510
  888 CONTINUE                                                          SAT08520
      CALL PLOAD(1,Z)                                                   SAT08530
      CALL TLOAD(1,Z)                                                   SAT08540
      DELSQ=DEL**2                                                      SAT08550
      TDLI=.5/DEL                                                       SAT08560
      TDEL=2.0*DEL                                                      SAT08570
      MNINIT=1                                                          SAT08580
      MNMAXC=MNMAX                                                      SAT08590
      DO 20 I=1,4                                                       SAT08600
      DO 20 J=1,4                                                       SAT08610
      UNIT(I,J)=0.0                                                     SAT08620
   20 IF(I.EQ.J) UNIT(I,J)=1.0                                          SAT08630
      NMAX=MAXM*KMAX2                                                   SAT08640
      DO 22 K=1,NMAX                                                    SAT08650
      DO 22 I=1,4                                                       SAT08660
      ZDOT(I,K)=0.0                                                     SAT08670
      ZO(I,K)=0.0                                                       SAT08680
      Z2(I,K)=0.0                                                       SAT08690
      Z3(I,K)=0.0                                                       SAT08700
   22 Z(I,K)=0.0                                                        SAT08710
      IF(IC.EQ.0) GO TO 834                                             SAT08720
      CALL INITL (Z,ZO,Z2,Z3,ZDOT)                                      SAT08730
      ACO=CHAR*SIGO/ELAST                                               SAT08740
      ACM=SIGO*TKN**3/CHAR                                              SAT08750
      DO 830 M=1,MNMAX                                                  SAT08760
      MM=(M-1)*KMAX2                                                    SAT08770
      WRITE(6,126) N(M)                                                 SAT08780
      WRITE(6,127)                                                      SAT08790
      DO 831 K=2,KMAX1                                                  SAT08800
      MK=K+MM                                                           SAT08810
      TU=ACO*ZO(1,MK)                                                   SAT08820
      TV=ACO*ZO(2,MK)                                                   SAT08830
      TW=ACO*ZO(3,MK)                                                   SAT08840
      TM=ACM*ZO(4,MK)                                                   SAT08850
      KK=K-1                                                            SAT08860
  831 WRITE(6,71) KK,TU,TV,TW,TM                                        SAT08870
      WRITE(6,129)                                                      SAT08880
      DO 830 K=2,KMAX1                                                  SAT08890
      ACD=CHAR*SIGO/(ELAST*TEEO)                                        SAT08900
      AMD=SIGO*TKN**3/(CHAR*TEEO)                                       SAT08910
      MK=K+MM                                                           SAT08920
      TU=ACD*ZDOT(1,MK)                                                 SAT08930
      TV=ACD*ZDOT(2,MK)                                                 SAT08940
```

```
      TW=ACD*ZDOT(3,MK)                                                 SAT08950
      TM=AMD*ZDOT(4,MK)                                                 SAT08960
      KK=K-1                                                            SAT08970
      WRITE(6,71) KK,TU,TV,TW,TM                                        SAT08980
      DO 830 I=1,4                                                      SAT08990
      Z(I,MK)=ZO(I,MK)+ZDOT(I,MK)*DELCAD                                SAT09000
      Z2(I,MK)=ZO(I,MK)-ZDOT(I,MK)*DELCAD                               SAT09010
  830 Z3(I,MK)=ZO(I,MK)-2.*ZDOT(I,MK)*DELOAD                            SAT09020
  834 CONTINUE                                                          SAT09030
      ALCAD=1.0                                                         SAT09040
      CALL IMPERF (PHIXB,PHITB)                                         SAT09050
      CALL FMATRX (P,X,ZO,Z2,Z3,DEE,DST)                                SAT09060
      LSTEP=1                                                           SAT09070
      LCHANG=0                                                          SAT09080
      ITR=1                                                             SAT09090
      ICORFL=0                                                          SAT09100
      IF(MNMAX.EQ.MAXM) ICORFL=1                                        SAT09110
      IPASS=0                                                           SAT09120
      ITTEST=0
  400 CALL XANDZ (P,DEE,DST,X,Z,ZO,Z2,Z3,ZDOT,IS,JS,ID,JD,PHIXB,PHITB)  SAT09130
      IF(ITRMAX.EQ.1) GO TO 50                                          SAT09140
      MNMAXO=MNMAX                                                      SAT09150
      IF(IPASS.LT.2) CALL MODES (IS,JS,ID,JD,P,X,ZO,Z2,Z3,DEE,DST)
      IF(NCONV.EQ.1) GO TO 50                                           SAT09170
      IF(ITR.LT.ITRMAX) GO TO 23                                        SAT09180
      GO TO 365                                                         SAT09190
   50 FL=LSTEP                                                          SAT09200
      FI=IPRINT                                                         SAT09210
      LI=LSTEP/IPRINT                                                   SAT09220
      FLI=LI                                                            SAT09230
      FT=FLI-FL/FI                                                      SAT09240
      IF(FT.EQ.0.) CALL OUTPUT(IMODE,P,DEE,DST,X,Z,ZO,Z2,Z3,ZDOT,IS,JS, SAT09250
     1ID,JD,PHIXB,PHITB)                                                SAT09260
      IF(LSTEP.GE.LSMAX) GO TO 360                                      SAT09300
      DO 65 MN=1,MNMAXO                                                 SAT09310
      DO 65 K=1,KMAX2                                                   SAT09320
      IK=K+(MN-1)*KMAX2                                                 SAT09330
      DO 65 I=1,4                                                       SAT09340
      ZN=3.0*(Z(I,IK)-ZO(I,IK))+Z2(I,IK)                                SAT09350
      Z3(I,IK)=Z2(I,IK)                                                 SAT09360
      Z2(I,IK)=ZO(I,IK)                                                 SAT09370
      ZO(I,IK)=Z(I,IK)                                                  SAT09380
   65 Z(I,IK)=ZN                                                        SAT09390
      ALCAD=1.0                                                         SAT09400
      LSTEP=LSTEP+1                                                     SAT09410
      ITR=1                                                             SAT09420
      GO TO 400                                                         SAT09430
   23 ITR=ITR+1                                                         SAT09440
```

```
      GO TO 400                                                         SAT09450
  360 WRITE(6,271)                                                      SAT09460
      WRITE(6,188)(AVB(M),M=1,MAXM)                                     SAT09470
      WRITE(6,189) ITRPR                                                SAT09480
      GO TO 500                                                         SAT09490
  365 IF(LSTEP.EQ.1) GO TO 367                                          SAT09500
      WRITE(6,266) ITRMAX,LSTEP,NC                                      SAT09510
      WRITE(6,188)(AVB(M),M=1,MAXM)                                     SAT09520
      GO TO 500                                                         SAT09530
  367 WRITE(6,273)                                                      SAT09540
      GO TO 500                                                         SAT09550
C***********************************************************************SAT09560
   71 FORMAT(20X,I3,4X,4E20.6)                                          SAT09570
  112 FORMAT(////17X,12H  STATION   20H      B STIFFNESS   20H      D STSAT09580
     1IFFNESS   20H       B PRIME         20H       D PRIME        //)  SAT09590
  126 FORMAT(////5X,29HTHE INITIAL CONDITIONS FOR N=I3,8H  FOLLOW//)    SAT09600
  127 FORMAT(19X,7HSTATION,3X,20H         U                20H       V  SAT09610
     1    20H         W               20H        M S              //)   SAT09620
  129 FORMAT(///19X,7HSTATION,3X,20H       U DOT           20H       V DOT  SAT09630
     1          20H        W DOT            20H        M S DOT       //) SAT09640
  188 FORMAT(////' THE MAXIMUM VBAR FOR EACH MODE IS'/10E11.4)          SAT09650
  189 FORMAT(/'  THE MAXIMUMUM NUMBER OF ITERATIONS TAKEN IS   ',I3)    SAT09660
  266 FORMAT(1H1,35H   THE SOLUTION DID NOT CONVERGE INI3,24H ITERATIONSSAT09670
     1 AT TIME STEPI5,21H.  END PROBLEM NUMBERI4,1H.)                   SAT09680
  271 FORMAT(1H1,79H          THE MAXIMUM NUMBER OF TIME STEPS HAVE BEENSAT09690
     1 TAKEN.  END PROBLEM NUMBERI4)                                    SAT09700
  273 FORMAT(1H1,69H          THE SOLUTION DID NOT CONVERGE FOR THE FIRSSAT09710
     1T TIME INCREMENT./11X,71HLOCK FOR AN ERROR IN THE INPUT DATA, OR TSAT09720
     2RY A SMALLER VALUE FOR DELOAD.)                                   SAT09730
  810 FORMAT(1H1, 5X,15H STATION        16H     RADIUS     16H     GAMMA SAT09740
     1      16H     OMEGA S     16H     OMEGA THETA16H     DEOMEGA S  16H SAT09750
     2      MASS        //)                                             SAT09760
  813 FORMAT( 8X,I3, 9X,6E16.4)                                         SAT09770
 8888 FORMAT (' ',T10,'EXECUTING IN SUBROUTINE "DYNAMIC"')              SAT09780
  500 RETURN                                                            SAT09790
      END                                                               SAT09800
      SUBROUTINE PLOTIT(X,Y,NN,MODCUR)                                  SAT09810
C***********************************************************************SAT09820
C     THIS SUBROUTINE AND THE THREE THAT FOLLOW IT COMPRISE THE SELF-  *SAT09830
C     CONTAINED PLOTTING CAPABILITY OF PROGRAM SATANS.  THEY RECEIVE   *SAT09840
C     DATA TO BE PLOTTED, ROUND IT, SCALE IT, AND DRAW IS ON THE HIGH- *SAT09850
C     SPEED LINE PRINTER.                                              *SAT09860
C***********************************************************************SAT09870
      DIMENSION X( 1), Y( 1), RANGE(4)                                  SAT09880
      EQUIVALENCE (RANGE(1),XMAX), (RANGE(2),XMIN),(RANGE(3),YMAX),     SAT09890
     1            (RANGE(4),YMIN)                                       SAT09900
      KN=IABS(NN)                                                       SAT09910
      IF(MODCUR.GT.1) GO TO 5                                           SAT09920
```

```
C**********************************************************************SAT09930
C     FIND MAX & MIN FOR SCALE COMPUTATIONS                            *SAT09940
C**********************************************************************SAT09950
      XMAX=-1.E20                                                       SAT09960
      XMIN=1.E20                                                        SAT09970
      YMAX=-1.E20                                                       SAT09980
      YMIN=1.E20                                                        SAT09990
      DO 1 I=1,KN                                                       SAT10000
      IF(X(I).LT.XMAX) GO TO 6                                          SAT10010
      XMAX=X(I)                                                         SAT10020
    6 IF(X(I).GT.XMIN) GO TO 7                                          SAT10030
      XMIN=X(I)                                                         SAT10040
    7 IF(Y(I).LT.YMAX) GO TO 8                                          SAT10050
      YMAX=Y(I)                                                         SAT10060
    8 IF(Y(I).GT.YMIN) GO TO 1                                          SAT10070
      YMIN=Y(I)                                                         SAT10080
    1 CONTINUE                                                          SAT10090
C**********************************************************************SAT10100
C     IF NOT AUTOSCALE GO TO CALL DRAWIT                               *SAT10110
C**********************************************************************SAT10120
      IF(NN.GT.0) GO TO 5                                               SAT10130
C**********************************************************************SAT10140
C     COMPUTE X-SCALE & NEW XMAX AND XMIN                              *SAT10150
C**********************************************************************SAT10160
      CALL SCALIT(XMAX,XMIN,4)                                          SAT10170
C**********************************************************************SAT10180
C     COMPUTE Y-SCALE & NEW YMAX AND YMIN                              *SAT10190
C**********************************************************************SAT10200
      CALL SCALIT(YMAX,YMIN,6)                                          SAT10210
C**********************************************************************SAT10220
C     PLOT CURVE                                                       *SAT10230
C**********************************************************************SAT10240
    5 CALL DRAWIT(X,Y,KN,RANGE,1,MODCUR)                                SAT10250
      IF(MODCUR. EQ.1.OR.MODCUR.EQ.2) RETURN                            SAT10260
C**********************************************************************SAT10270
C     PRINT SCALES WHEN LAST CURVE PLOTTED                             *SAT10280
C**********************************************************************SAT10290
      XS=(XMAX-XMIN)/80.                                                SAT10300
      YS=(YMAX-YMIN)/60.                                                SAT10310
      WRITE(6,100) XS,YS                                                SAT10320
  100 FORMAT(  15X,'X-SCALE:  "*"=',E10.3,' UNITS'//                    SAT10330
     1         15X,'Y-SCALE:  "*"=',E10.3,' UNITS')                     SAT10340
      RETURN                                                            SAT10350
      END                                                               SAT10360
C**********************************************************************SAT10370
      SUBROUTINE SCALIT(XMAX,XMIN,IDIV)                                 SAT10380
      DIV=IDIV                                                          SAT10390
C**********************************************************************SAT10400
```

```
C     RCUND MAXIMUM TO NEXT HIGHEST 2 SIG FIGS                            *SAT10410
C*************************************************************************SAT10420
      XMAX=AMAX1(0.,XMAX)                                                  SAT10430
      CALL RCUND(XMAX,IMX,FMX)                                             SAT10440
      IMX=IMX-1                                                            SAT10450
      FMX=FMX*10.                                                          SAT10460
    3 XMX=FMX*10.**IMX                                                     SAT10470
      IF(XMX.GE.XMAX) GC TO 2                                              SAT10480
      FMX=FMX+1.                                                           SAT10490
      IMM=FMX                                                              SAT10500
      FMX=IMM                                                              SAT10510
      GC TO 3                                                              SAT10520
C*************************************************************************SAT10530
C     RCUND MINIMUM TO NEXT LCWEST 2 SIG FIGS                             *SAT10540
C*************************************************************************SAT10550
    2 XMIN=AMIN1(0.,XMIN)                                                  SAT10560
      CALL RCUND(XMIN,IMN,FMN)                                             SAT10570
      IMN=IMN-1                                                            SAT10580
      FMN=FMN*10.                                                          SAT10590
   14 XMN=FMN*10.**IMN                                                     SAT10600
      IF(XMIN.GE.XMN) GC TO 11                                             SAT10610
      FMN=FMN-1.                                                           SAT10620
      IMM=FMN                                                              SAT10630
      FMN=IMM                                                              SAT10640
      GC TO 14                                                             SAT10650
C*************************************************************************SAT10660
C     RCUND MAX & MIN TC 1. OR .1 IF RANGE LARGE                          *SAT10670
C*************************************************************************SAT10680
   11 XSC=XMX-XMN                                                          SAT10690
      IM=C                                                                 SAT10700
      SM=1.                                                                SAT10710
    9 IF(XSC/CIV.LE.SM) GC TO 12                                           SAT10720
      IF(ABS(XMN).LT.SM.AND.ABS(XMN).GT.C.) XMN=SIGN(SM,XMN)               SAT10730
      IF(ABS(XMX).LT.SM.AND.ABS(XMX).GT.0.) XMX=SIGN(SM,XMX)               SAT10740
   12 IF(IM.GT.0) GO TO 19                                                 SAT10750
      SM=.1                                                                SAT10760
      IM=IM+1                                                              SAT10770
      GC TO 9                                                              SAT10780
C*************************************************************************SAT10790
C     RCUND RANGE (MAX-MIN) TC 2 SIG FIGS                                 *SAT10800
C*************************************************************************SAT10810
   19 XSC=XMX-XMN                                                          SAT10820
      CALL RCUND(XSC,ISIC,FACTX)                                           SAT10830
C*************************************************************************SAT10840
C     FINC FACTCR WHICH IS MULTIPLE OF IDIV                               *SAT10850
C*************************************************************************SAT10860
      FACTX=FACTX*10.                                                      SAT10870
      CFAC=FACTX                                                           SAT10880
```

```
      IS10=IS10-1                                                       SAT10890
      IFX=FACTX                                                         SAT10900
      FACTX=IFX                                                         SAT10910
   2C IF(MCD(IFX,IDIV).EQ.0.AND.FACTX.GE.OFAC) GO TC 10                 SAT10920
      IFX=IFX+1                                                         SAT10930
      FACTX=IFX                                                         SAT10940
      GC TO 20                                                          SAT10950
   1C IF(IDIV.GT.4) GO TC 15                                            SAT10960
C***********************************************************************SAT10970
C     IF X SCALE BETWEEN 8. AND 10., RCUND TO 10.                      *SAT10980
C***********************************************************************SAT10990
      FFX=ABS(FACTX/10.)                                                SAT11000
      IF(FFX.GT.8..AND.FFX.LT.10.) FFX=10.                              SAT11010
      IF(FACT X.LT.O.) FXX=-10.                                         SAT11020
      FACTX=FFX*10.                                                     SAT11030
   15 XSC=FACTX*10.**IS10                                               SAT11040
C***********************************************************************SAT11050
C     CCMPUTE NEW MAX & MIN FRCM RCUNDEC SCALE                         *SAT11060
C***********************************************************************SAT11070
      IF(XM N*XM X.NE.O.) GO TO 4                                       SAT11080
      IF(XM N.LT.O.) XMIN=-XSC                                          SAT11090
      IF(XM X.GT.C.) XMAX=XSC                                           SAT11100
      RETURN                                                            SAT11110
    4 XMAX=XSC+XMN                                                      SAT11120
      XMIN=XMN                                                          SAT11130
      RETURN                                                            SAT11140
      END                                                               SAT11150
C***********************************************************************SAT11160
      SUBROUTINE ROUND(ANUM,IS,FACT)                                    SAT11170
C***********************************************************************SAT11180
C     EXPRESS ANUM IN SCIENTIFIC NOTATICN WHERE                        *SAT11190
C           ANUM=FACT*10.**IS WHERE FACT IS BETWEEN 1. ANC 9.9         *SAT11200
C***********************************************************************SAT11210
      IF(ANUM.EQ.0.) GC TC 15                                           SAT11220
      BNUM=ANUM                                                         SAT11230
      IF(BNUM.LT.O.) BNUM=-BNUM                                         SAT11240
      IS=ALCG(BNUM)*.43429448                                           SAT11250
      FACT=BNUM/10.**IS                                                 SAT11260
C***********************************************************************SAT11270
C     FIND PCWER OF 10                                                 *SAT11280
C***********************************************************************SAT11290
      ICC=-3                                                            SAT11300
      R2=0                                                              SAT11310
      DC 10 II=1,5                                                      SAT11320
      ICC=ICC+1                                                         SAT11330
      R1=R2                                                             SAT11340
      R2=10.**(ICD+1)                                                   SAT11350
      IF(FACT.GE.R1.AND.FACT.LT.R2) GC TO 8                             SAT11360
```

```
   10 CONTINUE                                                          SAT11370
    8 FACT=FACT*10.**(-IDD)                                             SAT11380
      IS=IS+IDD                                                         SAT11390
C***********************************************************************SAT11400
C     ROUND MANTISSA TO 2 SIG FIGS                                     *SAT11410
C***********************************************************************SAT11420
      IFAC=FACT*10.+.05                                                 SAT11430
      FACT=IFAC                                                         SAT11440
      FACT=FACT/10.                                                     SAT11450
      IF(FACT.LT.10.) GO TO 20                                          SAT11460
C***********************************************************************SAT11470
C     SET TO 1 IF LESS THAN 10.                                        *SAT11480
C***********************************************************************SAT11490
      FACT=1.                                                           SAT11500
      IS=IS+1                                                           SAT11510
C***********************************************************************SAT11520
C     IF INPUT NEGATIVE, SET MANTISSA NEGATIVE                         *SAT11530
C***********************************************************************SAT11540
   20 IF(ANUM.LT.0.) FACT=-FACT                                         SAT11550
      RETURN                                                            SAT11560
C***********************************************************************SAT11570
C     SET TO 0. IF 0.                                                  *SAT11580
C***********************************************************************SAT11590
   15 FACT=0.                                                           SAT11600
      IS=0                                                              SAT11610
      RETURN                                                            SAT11620
      END                                                               SAT11630
C***********************************************************************SAT11640
      SUBROUTINE DRAWIT(X,Y,NDATA,RANGE,KKZ,MODCUR)                     SAT11650
      DIMENSION GRID(61,81),XSCALE(5),YSCALE(7)                         SAT11660
      DIMENSION X(1),Y(1),RANGE(4)                                      SAT11670
      INTEGER*2 GRID,BLANK,DOT,XCHAR(4)/1H+,1H.,1H*,1HX/                SAT11680
      DATA DOT,BLANK/Z4B40,Z4040/                                       SAT11690
      KDATA=NDATA*KKZ                                                   SAT11700
      IF(MODCUR.GT.1) GO TO 444                                         SAT11710
C***********************************************************************SAT11720
C     GRID IS THE MATRIX USED TO PLOT THE POINTS                       *SAT11730
C***********************************************************************SAT11740
      IERR=0                                                            SAT11750
      XMAX=RANGE(1)                                                     SAT11760
      XMIN=RANGE(2)                                                     SAT11770
      YMAX=RANGE(3)                                                     SAT11780
      YMIN=RANGE(4)                                                     SAT11790
C***********************************************************************SAT11800
C     CHECKING X AND Y POINTS AND PLOTTING THOSE OUT OF RANGE          *SAT11810
C     AT THE MARGIN                                                    *SAT11820
C***********************************************************************SAT11830
      DO 30 I=1,KDATA,KKZ                                               SAT11840
```

```
      IF(X(I).GT.XMAX.OR.X(I).LT.XMIN.OR.Y(I).GT.YMAX.OR.Y(I).LT.YMIN)  SAT11850
     1  IERR=IERR+1                                                       SAT11860
      IF(X(I).LE.XMAX) GC TC 205                                          SAT11870
      X  (I)=XMAX                                                         SAT11880
      GCTC 21C                                                            SAT11890
  2C5 IF(X(I).GE.XMIN) GO TO 210                                          SAT11900
      X  (I)=XMIN                                                         SAT11910
  21C IF(Y(I).LE.YMAX) GC TO 215                                          SAT11920
      Y  (I)=YMAX                                                         SAT11930
      GCTC 3C                                                             SAT11940
  215 IF(Y(I).GE.YMIN) GC TO 30                                           SAT11950
      Y  (I)=YMIN                                                         SAT11960
   3C CCNTINUE                                                            SAT11970
C************************************************************************SAT11980
C     PLCTTING X AND Y AXIS , IF NECESSARY                              *SAT11990
C************************************************************************SAT12000
      JERR=0                                                              SAT12010
      XRANGE=XMAX-XMIN                                                    SAT12020
      YRANGE=YMAX-YMIN                                                    SAT12030
      IF (YRANGE.NE.0.)  GO TO 298                                        SAT12040
      IF(YMIN.EQ.0.) GO TC 889                                            SAT12050
      YMIN=0.                                                             SAT12060
      YRANGE=YMAX                                                         SAT12070
      GC TO 299                                                           SAT12080
  298 IF (XRANGE.NE.0.)  GO TO 299                                        SAT12090
      IF(XMIN.EQ.0.) GO TC 887                                            SAT12100
      XMIN=0.                                                             SAT12110
      XRANGE=XMAX                                                         SAT12120
C************************************************************************SAT12130
C     BLANKING CUT MATRIX-(GRID)                                        *SAT12140
C************************************************************************SAT12150
  299 DC 300 I=1,61                                                       SAT12160
      DC 301 JJ=1,81                                                      SAT12170
301   GRIC(I,JJ)=BLANK                                                    SAT12180
  3CC CCNTINUE                                                            SAT12190
      IF(XMAX*XMIN.GE.0.) GO TO 222                                       SAT12200
      IYAXIS=80.*(-XMIN)/XRANGE+1.5                                       SAT12210
      CC 40 I=1,61                                                        SAT12220
   4C GRIC(I,IYAXIS)=DOT                                                  SAT12230
  222 IF(YMAX*YMIN.GE.0.) GO TO 333                                       SAT12240
      IXAXIS=60.*YMAX/YRANGE+1.5                                          SAT12250
      CC 60 I=1,81                                                        SAT12260
   6C GRID(IXAXIS,I)=DCT                                                  SAT12270
C************************************************************************SAT12280
C     CCMPUTE PROPER SCALE NUMBERS                                      *SAT12290
C************************************************************************SAT12300
  333 XINCR=XRANGE/4.                                                     SAT12310
      YINCR=YRANGE/6.                                                     SAT12320
```

```
      XSCALE(1)=XMAX                                                    SAT12330
      XSCALE(5)=XMIN                                                    SAT12340
      DO 80 I=2,4                                                       SAT12350
      XSCALE(I)=XSCALE(I-1)-XINCR                                       SAT12360
      IF(ABS(XSCALE(I)).LT.1.E-4) XSCALE(I)=0.                          SAT12370
   80 CONTINUE                                                          SAT12380
      YSCALE(1)=YMAX                                                    SAT12390
      YSCALE(7)=YMIN                                                    SAT12400
      DO 81 I=2,6                                                       SAT12410
      YSCALE(I)=YSCALE(I-1)-YINCR                                       SAT12420
      IF(ABS(YSCALE(I)).LT.1.E-4) YSCALE(I)=0.                          SAT12430
   81 CONTINUE                                                          SAT12440
      DO 85 II=1,2                                                      SAT12450
      JJ=6-II                                                           SAT12460
      XT=XSCALE(JJ)                                                     SAT12470
      XSCALE(JJ)=XSCALE(II)                                             SAT12480
   85 XSCALE(II)=XT                                                     SAT12490
C***********************************************************************SAT12500
C***********************************************************************SAT12510
  444 IF(MODCUR.LT.2) JSET=0                                            SAT12520
      IF(JERR.GT.0) GO TO 885                                           SAT12530
      JSET=JSET+1                                                       SAT12540
      IF(JSET.GT.4) JSET=1                                              SAT12550
      DO 700 I=1,KDATA,KKZ                                              SAT12560
      IPTX=60.*(YMAX-Y(I))/YRANGE+1.5                                   SAT12570
      IPTY=80.*(X(I)-XMIN)/XRANGE+1.5                                   SAT12580
      IF(IPTX.GT.61.OR.IPTY.GT.81) GO TO 70                             SAT12590
      IF(IPTX.LE.0.OR.IPTY.LE.0)GO TO 70                                SAT12600
      GRID(IPTX,IPTY) = XCHAR(JSET)                                     SAT12610
      GO TO 700                                                         SAT12620
   70 IERR=IERR+1                                                       SAT12630
  700 CONTINUE                                                          SAT12640
C***********************************************************************SAT12650
C     OUTPUT SECTION WITH GRAPH                                        *SAT12660
C***********************************************************************SAT12670
      IF(MODCUR.EQ.1.OR.MODCUR.EQ.2) RETURN                             SAT12680
      AXR=ABS(XRANGE)                                                   SAT12690
      AYR=ABS(YRANGE)                                                   SAT12700
      IF(AXR.LT.1.E+8.AND.AXR.GE..95) GO TO 400                         SAT12710
      WRITE(6,17) XSCALE                                                SAT12720
   17 FORMAT(12X,1PE10.3,4(10X,E10.3)/15X,'**',8('+*********'),'+**')   SAT12730
      GO TO 401                                                         SAT12740
  400 WRITE(6,117) XSCALE                                               SAT12750
  117 FORMAT( 8X,F11.2,4( 9X,F11.2)/15X,'**',8('+*********'),'+**')     SAT12760
  401 II=1                                                              SAT12770
      DO 101 IK=1,61                                                    SAT12780
      IF(MOD(IK-1,10).NE.0) GO TO 92                                    SAT12790
      IF(AYR.LT.1.E+8.AND.AYR.GE..95) GO TO 404                         SAT12800
```

```
      WRITE(6,18) YSCALE(II),(GRID(IK,IX),IX=1,81),YSCALE(II)           SAT12810
   18 FORMAT(3X,1PE10.3,2X,1H+,1X,81A1,1X,1H+,2X,E10.3)                 SAT12820
      GO TO 405                                                         SAT12830
  404 WRITE(6,118) YSCALE(II),(GRID(IK,IX),IX=1,81),YSCALE(II)          SAT12840
  118 FORMAT(2X,F11.2,'  + ',81A1,' +  ',F11.2)                          SAT12850
  405 II=II+1                                                           SAT12860
      GO TO 101                                                         SAT12870
   92 WRITE(6,19) (GRID(IK,IX),IX=1,81)                                 SAT12880
   19 FORMAT(15X,'* ',81A1,' *')                                        SAT12890
  101 CONTINUE                                                          SAT12900
      IF(AXR.LT.1.E+8.AND.AXR.GE..95) GO TO 402                         SAT12910
      WRITE(6,22) XSCALE                                                SAT12920
   22 FORMAT(15X,'**',8('+*********'),'+**'/12X,1PE10.3,4(10X,E10.3),//)SAT12930
      GO TO 403                                                         SAT12940
  402 WRITE(6,217) XSCALE                                               SAT12950
  217 FORMAT(15X,'**',8('+*********'),'+**'/ 8X,F11.2,4( 9X,F11.2),//)  SAT12960
  403 IF(IERR.GT.0) WRITE(6,20) IERR                                    SAT12970
   20 FORMAT(10X 'NUMBER OF POINTS OUT OF RANGE =', I4)                 SAT12980
 1000 RETURN                                                            SAT12990
  889 WRITE(6,888)                                                      SAT13000
  888 FORMAT(' ALL Y VALUES=0. CANNOT SETUP PLOT GRID. CHECK MAX & MIN YSAT13010
     1 WHEN MODCUR=0 OR 1.')                                            SAT13020
      JERR=10                                                           SAT13030
      RETURN                                                            SAT13040
  887 WRITE(6,886)                                                      SAT13050
  886 FORMAT(' ALL X VALUES=0. CANNOT SETUP PLOT GRID. CHECK MAX & MIN XSAT13060
     1 WHEN MODCUR=0 OR 1.')                                            SAT13070
      JERR=10                                                           SAT13080
      RETURN                                                            SAT13090
  885 WRITE(6,884)                                                      SAT13100
  884 FORMAT(' GRID NOT SETUP WHEN MODCUR LAST 0 OR 1. NO PLOT UNTIL GRISAT13110
     1D PROPERLY SETUP')                                                SAT13120
      RETURN                                                            SAT13130
      END                                                               SAT13140
      SUBROUTINE OUTPUT(IMODE,P,DEE,DST,X,Z,Z0,Z2,Z3,ZDOT,IS,JS,ID,JD,  SAT13150
     1PHIXB,PHITB)                                                      SAT13160
C***********************************************************************SAT13170
C          THIS SUBROUTINE PREPARES THE PRINTOUT MATERIAL.  EVERY IPRINT *SAT13180
C          CONVERGED SOLUTION IS PRINTED.  THE FOURIER COEFFICIENTS OF THESAT13190
C          INPLANE FORCES, MERIDIONAL TRANSVERSE FORCE, CIRCUMFERENTIAL  *SAT13200
C          BENDING MOMENT, TWISTING MOMENT AND ROTATIONS CAN BE COMPUTED *SAT13210
C          AND PRINTED WITH THE SOLUTION Z FOR THE FOURIER COEFFICIENTS  *SAT13220
C          OF THE THREE DISPLACEMENTS AND MERIDIONAL BENDING MOMENT.  THISSAT13230
C          OUTPUT MATERIAL IS CONVERTED FROM DIMENSIONLESS FORM TO DIMEN-*SAT13240
C          SIONAL FORM HERE.  PROVISION IS MADE TO PRINT ONLY AT STATIONS*SAT13250
C          1, IFREQ+1, 2IFREQ+1, ETC.  THIS SUBROUTINE ALSO PERFORMS THE *SAT13260
C          SUMMATION PROCESS FOR COMPUTING THE TOTAL VALUES OF THE FORCES,SAT13270
C          MOMENTS, DISPLACEMENTS AND ROTATIONS AT THE NTHMAX POSITIONS  *SAT13280
```

```
C        ARCUND THE CIRCUMFERENCE PRESCRIBED IN THE INPUT CATA.              *
C     CATA TO BE PLOTTED IS PREPARED HERE, IF REQ'C                          *
C*****************************************************************************
      IMPLICIT LCGICAL*1 ($)
      REAL NU,MT,MX,MTH,MXT,MTS,KX,KT,KXT,LAM,LAM2,MASS
      DIMENSION P(4,4,1),DEE(4,4,1),DST(4,4,1),X(4,1),Z(4,1),
     1ZC(4,1),Z2(4,1),Z3(4,1),ZDOT(4,1),IS(99,1),JS(99,1),IC(99,1),
     2JC(99,1),PHIXB(1),PHITB(1)
      CCMMON /IBL2/ N(99),MNINIT
      CCMMON /IBL3/ MO,M1,M2,M3
      CCMMON /IBL4/ KMAX,KL
      CCMMON /IBL5/ IBCINL,IBCFNL
      CCMMON /IBL7/ MNMAXD,MAXD(99),MAXS(99),MAXSY(99),IJS(99)
      CCMMON /IBL8/ LSTEP,ITR
      CCMMCN /IBL10/ IFREQ,NTHMAX
      CCMMCN /IBL12/ KMAX1,KMAX2,NCONV
      CCMMON /IBL13/ ITRMAX,LSMAX
     1/IBLJ/ JUMP
      CCMMCN /BL4/   ZF1M(4,4,99),ZF2M(4,4,99),
     1               ZF3M(4,4,99),ZF4M(4,4,99)
      CCMMON /BL5/   TT(99),MT(99),DT(99),DMT(99)
      CCMMON /BL6/   SOE,OSE,ALOAD
      CCMMON /BL7/   D1,S1
      CCMMON /BL8/   R(500),GAM(500),OMT(500)
      CCMMON /BL10/  PHIX(99),PHIT(99),PHI(99)
      CCMMON /BL11/  OMXI(500),PHEE,TO,T2
      CCMMON /BL12/  TDLI,TDEL
      CCMMCN /BL14/  LAM2,LSD18,LSD1N
      CCMMCN /BL15/  NU,U1(99),V1(99),W1(99),V2(99),U2(99),W2(99),U3(99),
     1               V3(99),W3(99)
      CCMMON /BL17/  DEL
      CCMMCN /BL19/  TH(36)
      CCMMON /BL20/  DEDMX(500)
      CCMMON /BL27/  BX3(99),BT3(99),BXT3(99),BE3(99)
      CCMMCN /BL31/  DELSG,EXT1(99)
      CCMMCN /BL32/  TKN,ELAST,CHAR,SIGC
      CCMMON /BL100/ TEEC,$DYNMC
      CCMMON /BL101/ DELSD
      CCMMON /BL102/ DELCAD
      CCMMON /BL103/ MASS(500)
      CCMMON /BL110/ TX(99),TTH(99),TXT(99),MX(99),MTH(99),MXT(99),
     1               QS(99)
      CCMMCN /BL111/ ABZ,ABZC,ABZN,ABZ3,CC2
      CCMMON /BLPLOT/  IRADII,IGAMMA,ICMEGS,IOMEGT,ICECMS,IBSTIF,IDSTIF,
     1                 IBBSTF,IDDSTF,IPR,IPS,IPT,ITT,IMT,ICTT,ICMT,INS,
     2                 INTH,INSTH,IQS,IMS,IMTH,IMSTH,IU,IV,IW,IFHIS,
     3                 IPHIT,IPHI,$PLOTS,$MODAL
      CCMMCN /BLPLT1/  XRADII(200),YGAMMA(200),YCMEGS(200),YCMEGT(200),
```

```
     1                YDECMS(200),YBSTIF(200),YCSTIF(200),YBBSTF(200),   SAT13770
     2                YDDSTF(200),YPR(200),YPS(200),YPT(200),YTT(200),   SAT13780
     3                YMT(200),YDTT(200),YDMT(200),YNS(200),YNTH(200),   SAT13790
     4                YNSTH(200),YQS(200),YMS(200),YMTH(200),YMSTH(200),SAT13800
     5                YU(200),YV(200),YW(200),YPHIS(200),YPHIT(200),     SAT13810
     6                YPHI(200),XSTATN(200)                              SAT13820
      DIMENSION PTF(500),PF(500)                                         SAT13830
C***********************************************************************SAT13840
      AEZC=SIGO/ELAST                                                    SAT13850
      IF ($DYNMC) GO TO 181                                              SAT13860
      WRITE(6,101) LSTEP,ALOAD,ITR                                       SAT13870
      GO TO 182                                                          SAT13880
  181 TI=LSTEP*DELOAD                                                    SAT13890
      DTI=TI*TEEO                                                        SAT13900
      WRITE(6,151) LSTEP,TI,DTI,ITR                                      SAT13910
  182 LAM=TKN/CHAR                                                       SAT13920
      ENL=1.                                                             SAT13930
      AEZ=SIGO*TKN                                                       SAT13940
      AEZ3=AEZ*TKN*TKN/CHAR                                              SAT13950
      AEZN=CHAR*SIGO/ELAST                                               SAT13960
      IF(ITRMAX.EQ.1) ENL=0.                                             SAT13970
      CC2=1.-NU**2                                                       SAT13980
      C2I=1./CD2                                                         SAT13990
      CPI=1./S1                                                          SAT14000
      CNI=1./C1                                                          SAT14010
      TCLSQI=.5/CELSQ                                                    SAT14020
      ICHCK1=IABS(INS)+IABS(INTH)+IABS(INSTH)+IABS(IQS)+IABS(IMS)        SAT14030
     1       +IABS(IMTH)+IABS(IMSTH)                                     SAT14040
      ICHCK2=IABS(IU)+IABS(IV)+IABS(IW)+IABS(IPHIS)+IABS(IPHIT)          SAT14050
     1       +IABS(IPHI)                                                 SAT14060
      IF(NTHMAX.EQ.0) GO TO 991                                          SAT14070
      DO 21 NTH=1,NTHMAX                                                 SAT14080
      DO 1 MN=1,MNMAXO                                                   SAT14090
      I1=1+(MN-1)*KMAX2                                                  SAT14100
      I2=I1+1                                                            SAT14110
      U1(MN)=Z(1,I1)                                                     SAT14120
      U2(MN)=Z(1,I2)                                                     SAT14130
      V1(MN)=Z(2,I1)                                                     SAT14140
      V2(MN)=Z(2,I2)                                                     SAT14150
      W1(MN)=Z(3,I1)                                                     SAT14160
    1 W2(MN)=Z(3,I2)                                                     SAT14170
      THET=TH(NTH)                                                       SAT14180
      WRITE(6,116) THET                                                  SAT14190
      DO 121 K=1,KMAX                                                    SAT14200
      K1=K+1                                                             SAT14210
      CALL BCB(K,BS,DB,CS,DC)                                            SAT14220
      IF(K.EQ.1.AND.IBCINL.LT.0) CALL PCLE(K,P,DEE,CST,X,Z,ZO,Z2,Z3,     SAT14230
     1ZCCT,IS,JS,ID,JD,PHIXB,PHITB)                                      SAT14240
```

```
      IF(K.EQ.1.AND.IBCINL.LT.0) GO TO 999
      IF(K.EQ.KMAX.AND.IBCFNL.LT.0) CALL POLE(K,P,CEE,DST,X,Z,ZO,Z2,Z3,
     1ZCCT,IS,JS,ID,JD,PHIXB,PHITB)
      IF(K.EQ.KMAX.AND.IBCFNL.LT.0) GO TO 999
      CALL PHIBET(K,Z,IS,JS,ID,JD,PHIXB,PHITB)
      DEX=DEGMX(K)
      RRA=1./R(K)
      OX=CMXI(K)
      OT=CMT(K)
      GA=GAM(K)
      DOXT=OX-OT
      GDO=GA*DOXT
      DD2D=DD2*DS
      DO 3 MN=1,MNMAXO
      EN=N(MN)
      ENR=EN*RRA
      CALL TLOAD(K,Z)
      TTS=TT(MN)*ALOAD
      EX=(U3(MN)-U1(MN))*TDLI+OX*W2(MN)+ENL*OSE*(BX3(MN)+BE3(MN))
      ET=ENR*V2(MN)+GA*U2(MN)+OT*W2(MN)+ENL*OSE*(BT3(MN)+BE3(MN))
      EXT=.5*((V3(MN)-V1(MN))*TDLI-ENR*U2(MN)-GA*V2(MN)+ENL*SCE*BXT3(MN)
     1)
      KT=ENR*PHIT(MN)+GA*PHIX(MN)
      KXT=.5*(ENR*(-PHIX(MN)-GA*W2(MN)+(W3(MN)-W1(MN))*TDLI)+GDO*V2(MN)
     1     +OT*(V3(MN)-V1(MN))*TDLI-GA*PHIT(MN)-DOXT*PHI(MN))
      TX(MN)=ES*(EX+NU*ET)-TTS
      TTH(MN)=BS*(ET+NU*EX)-TTS
      TXT(MN)=BS*D1*EXT
      MK1=K1+(MN-1)*KMAX2
      MX(MN)=Z(4,MK1)
      MTH(MN)=NU*MX(MN)+DD2D*KT-D1*MT(MN)*ALOAD
      MXT(MN)=DS*D1*KXT
      MK11=MK1+1
      MKK1=MK1-1
      CS(MN)=SIGO*TKN*LAM2*(GA*MX(MN)+(Z(4,MK11)-Z(4,MKK1))*TDLI
     1      +ENR*MXT(MN)-GA*MTH(MN))
      MX(MN)=MX(MN)*ABZ3
      MTH(MN)=MTH(MN)*ABZ3
      MXT(MN)=MXT(MN)*ABZ3
      TX(MN)=TX(MN)*ABZ
      TTH(MN)=TTH(MN)*ABZ
      TXT(MN)=TXT(MN)*ABZ
      PHIX(MN)=PHIX(MN)*ABZO
      PHIT(MN)=PHIT(MN)*ABZO
      PHI(MN)=PHI(MN)*ABZO
      U1(MN)=U2(MN)
      U2(MN)=U3(MN)
      V1(MN)=V2(MN)
```

```
      V2(MN)=V3(MN)                                                     SAT14730
      W1(MN)=W2(MN)                                                     SAT14740
    3 W2(MN)=W3(MN)                                                     SAT14750
      FK=K-1                                                            SAT14760
      FIFREQ=IFREQ                                                      SAT14770
      KTST=(K-1)/IFREQ                                                  SAT14780
      FKTST=KTST                                                        SAT14790
      FKTEST=FK/FIFREQ-FKTST                                            SAT14800
      IF(K.EQ.1.OR.K.EQ.KMAX) GO TO 999                                 SAT14810
      IF(FKTEST.NE.0.) GO TO 2                                          SAT14820
  999 X(1,K)=0.                                                         SAT14830
      X(2,K)=0.                                                         SAT14840
      X(3,K)=0.                                                         SAT14850
      X(4,K)=0.                                                         SAT14860
      PTF(K)=0.                                                         SAT14870
      PF(K)=0.                                                          SAT14880
      AMX=0.                                                            SAT14890
      AMTH=0.                                                           SAT14900
      AMXTH=0.                                                          SAT14910
      ANX=0.                                                            SAT14920
      ANTH=0.                                                           SAT14930
      ANXTH=0.                                                          SAT14940
      AQS=0.                                                            SAT14950
      IF(JUMP.EQ.2) GO TO 73                                            SAT14960
      DO 72 MN=1,MNMAXO                                                 SAT14970
      EN=N(MN)                                                          SAT14980
      FC=EN*THET                                                        SAT14990
      SN=SIN(FC)                                                        SAT15000
      CS=COS(FC)                                                        SAT15010
      X(1,K)=X(1,K)+U1(MN)*CS*ABZN                                      SAT15020
      X(2,K)=X(2,K)+V1(MN)*SN*ABZN                                      SAT15030
      X(3,K)=X(3,K)+W1(MN)*CS*ABZN                                      SAT15040
      X(4,K)=X(4,K)+PHIX(MN)*CS                                         SAT15050
      PTF(K)=PTF(K)+PHIT(MN)*SN                                         SAT15060
      AMX=AMX+MX(MN)*CS                                                 SAT15070
      AMTH=AMTH+MTH(MN)*CS                                              SAT15080
      AMXTH=AMXTH+MXT(MN)*SN                                            SAT15090
      ANX=ANX+TX(MN)*CS                                                 SAT15100
      ANTH=ANTH+TTH(MN)*CS                                              SAT15110
      AQS=AQS+QS(MN)*CS                                                 SAT15120
      ANXTH=ANXTH+TXT(MN)*SN                                            SAT15130
   72 PF(K)=PF(K)+PHI(MN)*SN                                            SAT15140
C***********************************************************************SAT15150
      GO TO 429                                                         SAT15160
   73 DO 74 MN=3,MNMAXO,JUMP                                            SAT15170
      EN=N(MN)                                                          SAT15180
      FC=EN*THET                                                        SAT15190
      SN=SIN(FC)                                                        SAT15200
```

```
      CS=COS(FC)                                                        SAT15210
      MNM=MN-1                                                          SAT15220
      X(1,K)=X(1,K)+(U1(MN)*CS+U1(MNM)*SN)*ABZN                         SAT15230
      X(2,K)=X(2,K)+(V1(MN)*SN+V1(MNM)*CS)*ABZN                         SAT15240
      X(3,K)=X(3,K)+(W1(MN)*CS+W1(MNM)*SN)*ABZN                         SAT15250
      X(4,K)=X(4,K)+PHIX(MN)*CS+PHIX(MNM)*SN                            SAT15260
      PTF(K)=PTF(K)+PHIT(MN)*SN+PHIT(MNM)*CS                            SAT15270
      AMX=AMX+MX(MN)*CS+MX(MNM)*SN                                      SAT15280
      AMTH=AMTH+MTH(MN)*CS+MTH(MNM)*SN                                  SAT15290
      AMXTH=AMXTH+MXT(MN)*SN+MXT(MNM)*CS                                SAT15300
      ANX=ANX+TX(MN)*CS+TX(MNM)*SN                                      SAT15310
      ANTH=ANTH+TTH(MN)*CS+TTH(MNM)*SN                                  SAT15320
      ANXTH=ANXTH+TXT(MN)*SN+TXT(MNM)*CS                                SAT15330
      AQS=AQS+QS(MN)*CS+QS(MNM)*SN                                      SAT15340
   74 PF(K)=PF(K)+PHI(MN)*SN+PHI(MNM)*CS                                SAT15350
      X(1,K)=X(1,K)+U1(1)*ABZN                                          SAT15360
      X(2,K)=X(2,K)+V1(1)*ABZN                                          SAT15370
      X(3,K)=X(3,K)+W1(1)*ABZN                                          SAT15380
      X(4,K)=X(4,K)+PHIX(1)                                             SAT15390
      PTF(K)=PTF(K)+PHIT(1)                                             SAT15400
      PF(K)=PF(K)+PHI(1)                                                SAT15410
      AMX=AMX+MX(1)                                                     SAT15420
      AMTH=AMTH+MTH(1)                                                  SAT15430
      AMXTH=AMXTH+MXT(1)                                                SAT15440
      ANX=ANX+TX(1)                                                     SAT15450
      ANTH=ANTH+TTH(1)                                                  SAT15460
      ANXTH=ANXTH+TXT(1)                                                SAT15470
      AQS=AQS+QS(1)                                                     SAT15480
C***********************************************************************SAT15490
  429 CONTINUE                                                          SAT15500
      IF(K.EQ.1) WRITE(6,117)                                           SAT15510
      WRITE(6,118) K,ANX,ANTH,ANXTH,AQS,AMX,AMTH,AMXTH                  SAT15520
      IF ($MOCAL.OR.(ICHCK1.EQ.0)) GO TO 2                              SAT15530
      YNS(K)=ANX                                                        SAT15540
      YNTH(K)=ANTH                                                      SAT15550
      YNSTH(K)=ANXTH                                                    SAT15560
      YQS(K)=AQS                                                        SAT15570
      YMS(K)=AMX                                                        SAT15580
      YMTH(K)=AMTH                                                      SAT15590
      YMSTH(K)=AMXTH                                                    SAT15600
    2 CONTINUE                                                          SAT15610
  121 CONTINUE                                                          SAT15620
      DO 660 K=1,KMAX                                                   SAT15630
      FK=K-1                                                            SAT15640
      FIFREQ=IFREQ                                                      SAT15650
      KTST=(K-1)/IFREQ                                                  SAT15660
      FKTST=KTST                                                        SAT15670
      FKTEST=FK/FIFREQ-FKTST                                            SAT15680
```

```
      IF(K.EQ.1.OR.K.EQ.KMAX) GO TO 661
      IF(FKTEST.NE.0.) GO TO 658
  661 IF(K.EQ.1) WRITE(6,217)
      WRITE(6,218) K,X(1,K),X(2,K),X(3,K),X(4,K),PTF(K),PF(K)
      IF ($MODAL.OR.(ICHCK2.EQ.0)) GO TO 658
      YU(K)=X(1,K)
      YV(K)=X(2,K)
      YW(K)=X(3,K)
      YPHIS(K)=X(4,K)
      YPHIT(K)=PTF(K)
      YPHI(K)=PF(K)
  658 DO 659 I=1,4
  659 X(I,K)=0.
  660 CONTINUE
      IF ($PLOTS.AND..NOT.$MODAL.AND.((ICHCK1.GT.0).OR.(ICHCK2.GT.0)))
     1   CALL PLOT2(NTH)
   21 CONTINUE
  551 IF(IMODE.LE.0) RETURN
      DO 534 MN=1,MNMAXO
      WRITE(6,749) N(MN)
      DO 521 MM=1,MNMAXO
      I1=1+(MM-1)*KMAX2
      I2=I1+1
      U1(MM)=Z(1,I1)
      U2(MM)=Z(1,I2)
      V1(MM)=Z(2,I1)
      V2(MM)=Z(2,I2)
      W1(MM)=Z(3,I1)
      W2(MM)=Z(3,I2)
  521 CONTINUE
      DO 445 K=1, KMAX
      K1=K+1
      CALL BCB(K,BS,DB,CS,DC)
      IF(K.EQ.1.AND.IBCINL.LT.0) CALL POLE(K,P,DEE,DST,X,Z,ZO,Z2,Z3,
     1ZDCT,IS,JS,ID,JD,PHIXB,PHITB)
      IF(K.EQ.KMAX.AND.IBCFNL.LT.0) CALL POLE(K,P,DEE,DST,X,Z,ZO,Z2,Z3,
     1ZDCT,IS,JS,ID,JD,PHIXB,PHITB)
      TXZ=TX(MN)
      TTHZ=TTH(MN)
      TXTZ=TXT(MN)
      AMXZ=MX(MN)
      AMTHZ=MTH(MN)
      AMXTZ=MXT(MN)
      QSZ=QS(MN)
      X(1,K)=PHIX(MN)
      X(2,K)=PHIT(MN)
      X(3,K)=PHI(MN)
      IF(K.EQ.1.AND.IBCINL.LT.0) GO TO 583
```

```
      IF(K.EQ.KMAX.AND.IBCFNL.LT.0) GC TC 583                          SAT16170
      CALL PHIBET(K,Z,IS,JS,ID,JD,PHIXB,PHITB)                         SAT16180
      DEX=DECMX(K)                                                     SAT16190
      RRA=1./R(K)                                                      SAT16200
      OX=CMXI(K)                                                       SAT16210
      CT=CMT(K)                                                        SAT16220
      GA=GAM(K)                                                        SAT16230
      DCXT=OX-CT                                                       SAT16240
      GCC=GA*DCXT                                                      SAT16250
      DD2D=DD2*DS                                                      SAT16260
      EN=N(MN)                                                         SAT16270
      ENR=EN*RRA                                                       SAT16280
      CALL TLCAD(K,Z)                                                  SAT16290
      TTS=TT(MN)*ALOAD                                                 SAT16300
      EX=(U3(MN)-U1(MN))*TDLI+OX*W2(MN)+ENL*OSE*(BX3(MN)+BE3(MN))      SAT16310
      ET=ENR*V2(MN)+GA*U2(MN)+OT*W2(MN)+ENL*OSE*(BT3(MN)+BE3(MN))      SAT16320
      EXT=.5*((V3(MN)-V1(MN))*TDLI-ENR*U2(MN)-GA*V2(MN)+ENL*SOE*BXT3(MN)SAT16330
     1)                                                                SAT16340
      KT=ENR*PHIT(MN)+GA*PHIX(MN)                                      SAT16350
      KXT=.5*(ENR*(-PHIX(MN)-GA*W2(MN)+(W3(MN)-W1(MN))*TDLI)+GDO*V2(MN) SAT16360
     1     +OT*(V3(MN)-V1(MN))*TDLI-GA*PHIT(MN)-DCXT*PHI(MN))          SAT16370
      TXZ=(BS*(EX+NU*ET)-TTS)*ABZ                                      SAT16380
      TTHZ=(BS*(ET+NU*EX)-TTS)*ABZ                                     SAT16390
      TXTZ=BS*D1*EXT*ABZ                                               SAT16400
      MK1=K1+(MN-1)*KMAX2                                              SAT16410
      AMXZ=Z(4,MK1)                                                    SAT16420
      AMTHZ=NU*AMXZ+DD2D*KT-D1*MT(MN)*ALCAD                            SAT16430
      AMXTZ=DS*C1*KXT                                                  SAT16440
      MK11=MK1+1                                                       SAT16450
      MKK1=MK1-1                                                       SAT16460
      CS2=SIGO*TKN*LAM2*(GA*AMXZ+(Z(4,MK11)-Z(4,MKK1))*TCLI+ENR*AMXTZ  SAT16470
     1     -GA*AMTHZ)                                                  SAT16480
      AMXZ=AMXZ*ABZ3                                                   SAT16490
      AMTHZ=AMTHZ*ABZ3                                                 SAT16500
      AMXTZ=AMXTZ*ABZ3                                                 SAT16510
      X(1,K)=PHIX(MN)*ABZC                                             SAT16520
      X(2,K)=PHIT(MN)*ABZC                                             SAT16530
      X(3,K)=PHI(MN)*ABZC                                              SAT16540
      DC 533 MM=1,MNMAXC                                               SAT16550
      U1(MM)=U2(MM)                                                    SAT16560
      U2(MM)=U3(MM)                                                    SAT16570
      V1(MM)=V2(MM)                                                    SAT16580
      V2(MM)=V3(MM)                                                    SAT16590
      W1(MM)=W2(MM)                                                    SAT16600
  533 W2(MM)=W3(MM)                                                    SAT16610
      FK=K-1                                                           SAT16620
      FIFREC=IFREQ                                                     SAT16630
      KTST=(K-1)/IFREQ                                                 SAT16640
```

```
      FKTST=KTST                                                        SAT16650
      FKTEST=FK/FIFREQ-FKTST                                            SAT16660
      IF(K.EQ.1.OR.K.EQ.KMAX) GO TO 583                                 SAT16670
      IF(FKTEST.NE.0.) GO TO 445                                        SAT16680
  583 CONTINUE                                                          SAT16690
      IF(K.EQ.1) WRITE(6,117)                                           SAT16700
      WRITE(6,118) K,TXZ,TTHZ,TXTZ,QSZ,AMXZ,AMTHZ,AMXTZ                 SAT16710
      IF (.NOT.$PLOTS.OR..NOT.$MODAL.OR.(ICHCK1.EQ.0)) GO TO 445        SAT16720
      YNS(K)=TXZ                                                        SAT16730
      YNTH(K)=TTHZ                                                      SAT16740
      YNSTH(K)=TXTZ                                                     SAT16750
      YQS(K)=QSZ                                                        SAT16760
      YMS(K)=AMXZ                                                       SAT16770
      YMTH(K)=AMTHZ                                                     SAT16780
      YMSTH(K)=AMXTZ                                                    SAT16790
  445 CONTINUE                                                          SAT16800
  446 WRITE(6,217)                                                      SAT16810
      DO 447 K=1,KMAX                                                   SAT16820
      FK=K-1                                                            SAT16830
      FIFREQ=IFREQ                                                      SAT16840
      KTST=(K-1)/IFREQ                                                  SAT16850
      FKTST=KTST                                                        SAT16860
      FKTEST=FK/FIFREQ-FKTST                                            SAT16870
      IF(K.EQ.1.OR.K.EQ.KMAX) GO TO 593                                 SAT16880
      IF(FKTEST.NE.0.) GO TO 447                                        SAT16890
  593 KZ=K+1+(MN-1)*KMAX2                                               SAT16900
      UP=Z(1,KZ)*ABZN                                                   SAT16910
      VP=Z(2,KZ)*ABZN                                                   SAT16920
      WP=Z(3,KZ)*ABZN                                                   SAT16930
      WRITE(6,218) K,UP,VP,WP,X(1,K),X(2,K),X(3,K)                      SAT16940
      IF (.NOT.$PLOTS.OR..NOT.$MODAL.OR.(ICHCK2.EQ.0)) GO TO 447        SAT16950
      YU(K)=UP                                                          SAT16960
      YV(K)=VP                                                          SAT16970
      YW(K)=WP                                                          SAT16980
      YPHIS(K)=X(1,K)                                                   SAT16990
      YPHIT(K)=X(2,K)                                                   SAT17000
      YPHI(K)=X(3,K)                                                    SAT17010
  447 CONTINUE                                                          SAT17020
      IF ($PLOTS.AND.$MODAL .AND.((ICHCK1.GT.0).OR.(ICHCK2.GT.0)))      SAT17030
     1   CALL PLOT2(1)                                                  SAT17040
  534 CONTINUE                                                          SAT17050
C***********************************************************************SAT17060
  101 FORMAT('1','          THE LOAD STEP NUMBER IS ',I2,'          THE LSAT17070
     1OAD FACTOR IS ',E11.4,'          THE SOLUTION CONVERGED IN ',I2,' SAT17080
     2ITERATIONS'////)                                                  SAT17090
  116 FORMAT('0','          THE SUMMED FORCES, MOMENTS, DISPLACEMENTS ANSAT17100
     1D ROTATIONS FOLLOW FOR THETA =',E15.6///)                         SAT17110
  117 FORMAT (/' STATION       N S                N THETA         N STHETA   SAT17120
```

```
 1          Q S                   M S                  M THETA            M STHETA'//)SAT17130
 118  FORMAT(1X,I3,3X,7E16.4)                                                         SAT17140
 151  FORMAT('1','   THE TIME STEP NUMBER IS ',I4,'        THE TIME IS ',F5SAT17150
     1.2,' OR ',E9.3,' SECONDS        THE SOLUTION CONVERGED IN ',I2,' ITESAT17160
     3RATIONS'/////)                                                                  SAT17170
 217  FORMAT(//' STATION          U                    V                   W          SAT17180
     1           PHI S            PHI THETA          PHI'/)                           SAT17190
 218  FORMAT(1X,I3,3X,6E16.4)                                                         SAT17200
 749  FORMAT('1',40X,' MODAL OUTPUT FOR MODE N = ',I3,' FOLLOWS')                     SAT17210
C*************************************************************************************SAT17220
      RETURN                                                                          SAT17230
      END                                                                             SAT17240
      SUBROUTINE XANDZ (P,DEE,DST,X,Z,ZC,Z2,Z3,ZDOT,IS,JS,ID,JD,PHIXB,                SAT17250
     1PHITB)                                                                          SAT17260
C*************************************************************************************SAT17270
C          THIS SUBROUTINE COMPUTES THE X VECTOR USING THE P, PBAR, AND   *SAT17280
C          P-HAT MATRICES AND SOLVES FOR THE Z VECTOR FOR THE APPLIED AND*SAT17290
C          PSUEDO LOADS.  THE SUBROUTINES PHIBET(K) AND TEAETA(K) ARE     *SAT17300
C          CALLED AND THE PREVIOUS SOLUTION FOR Z, OR THE ESTIMATED VALUE*SAT17310
C          OF Z IS USED TO CALCULATE THE NON-LINEAR BETA AND ETA TERMS.   *SAT17320
C          THE MATRICES FFS AND FLS ARE THE VALUES OF SMALL-F AT THE      *SAT17330
C          INITIAL AND FINAL EDGES OF THE SHELL.  THE SUBROUTINE FORCE(K)*SAT17340
C          IS CALLED TO CALCULATE THE LOAD VECTOR GEE, AND THE X VECTOR ATSAT17350
C          EACH MERIDIAN STATION.  ONCE THE X VECTOR IS OBTAINED FOR ALL *SAT17360
C          MERIDIAN STATIONS AND THE SOLUTION FOR Z(K+1) GIVEN BY EQUATIONSAT17370
C          (31B) IS OBTAINED, AND THE SOLUTION FOR Z(I) AT ALL THE OTHER *SAT17380
C          MERIDIAN STATIONS DEFINED BY EQUATION (29B) IS OBTAINED.  THE *SAT17390
C          SOLUTION Z AT THE IMAGINARY STATION OFF THE INITIAL EDGE OF   *SAT17400
C          SHELL IS OBTAINED LAST.  THE TEST FOR CONVERGENCE OF THE SOLU-*SAT17410
C          TION      IS MADE AS Z IS COMPUTED.  THE SPECIAL CONDITIONS FOR*SAT17420
C          COMPUTING Z AT EITHER AN INITIAL OR A FINAL POLE ARE ALSO IN  *SAT17430
C          THIS ROUTINE.                                                  *SAT17440
C*************************************************************************************SAT17450
      IMPLICIT LOGICAL*1 ($)                                                          SAT17460
      REAL NU,LAM2,MT,MASS                                                            SAT17470
      DIMENSION P(4,4,1),DEE(4,4,1),DST(4,4,1),X(4,1),Z(4,1),                         SAT17480
     1ZC(4,1),Z2(4,1),Z3(4,1),ZDOT(4,1),IS(99,1),JS(99,1),ID(99,1),                   SAT17490
     2JD(99,1),PHIXB(1),PHITB(1)                                                      SAT17500
      COMMON /IBL1/ MNMAX                                                             SAT17510
      COMMON /IBL3/ MO,M1,M2,M3                                                       SAT17520
      COMMON /IBL4/ KMAX,KL                                                           SAT17530
      COMMON /IBL5/ IBCINL,IBCFNL                                                     SAT17540
      COMMON /IBL6/ KLL                                                               SAT17550
      COMMON /IBL7/ MNMAXO,MAXD(99),MAXS(99),MAXSY(99),IJS(99)                        SAT17560
      COMMON /IBL8/ LSTEP,ITR                                                         SAT17570
      COMMON /IBL12/ KMAX1,KMAX2,NCONV                                                SAT17580
      COMMON /IBL13/ ITRMAX,LSMAX                                                     SAT17590
      COMMON /BL1/  A(4,4),BEE(4,4),C(4,4)                                            SAT17600
```

```
      COMMON /BL3/  PR(99),PX(99),PT(99)                                SAT17610
      COMMON /BL4/  ZF1M(4,4,99),ZF2M(4,4,99),
     1              ZF3M(4,4,99),ZF4M(4,4,99)
      COMMON /BL5/  TT(99),MT(99),DT(99),DMT(99)                        SAT17640
      COMMON /BL6/  SOE,CSE,ALOAD                                       SAT17650
      COMMON /BL7/  D1,S1                                               SAT17660
      COMMON /BL8/  R(500),GAM(500),OMT(500)                            SAT17670
      COMMON /BL9/  FFS(4,99),ELLS(4),GEES(4,99)
      COMMON /BL14/ LAM2,LSD18,LSDIN                                    SAT17690
      COMMON /BL15/ NU,U1(99),V1(99),W1(99),V2(99),U2(99),W2(99),U3(99),SAT17700
     1              V3(99),W3(99)                                       SAT17710
      COMMON /BL16/ EPS                                                 SAT17720
      COMMON /BL18/ EL1(4),ELL(4)                                       SAT17730
      COMMON /BL27/ BX3(99),BT3(99),BXT3(99),BE3(99)                    SAT17740
      COMMON /BL28/ EXX3(99),ETT3(99),ETX3(99),EXT3(99),EX3(99),ET3(99) SAT17750
      COMMON /BL29/ BX1(99),BT1(99),BXT1(99),BE1(99),BX2(99),BT2(99),   SAT17760
     1              BXT2(99),BE2(99)                                    SAT17770
      COMMON /BL30/ EXX1(99),ETT1(99),ETX1(99),EX1(99),ET1(99),EXX2(99),SAT17780
     1              ETT2(99),ETX2(99),EXT2(99),EX2(99),ET2(99)          SAT17790
      COMMON /BL31/ DELSQ,EXT1(99)                                      SAT17800
      COMMON /BL100/ TEEC,$DYNMC                                        SAT17810
      COMMON /BL101/ DELSD                                              SAT17820
      COMMON /BL102/ DELOAD                                             SAT17830
      COMMON /BL103/ MASS(500)                                          SAT17840
      DIMENSION ELLS(4),FLS(4),ZT(4),IPIVOT(4),INDEX(4,2)               SAT17850
     1,CL0(4,4),CL1(4,4),CL2(4,4)                                       SAT17860
     2,TZMAX(4,99),ZDD(4)
      EQUIVALENCE (CL0(1),ZF1M(1)),(CL1(1),ZF2M(1)),(CL2(1),ZF3M(1))    SAT17880
C***********************************************************************SAT17890
      DO 201 I=1,4                                                      SAT17900
      DO 201 M=1,MNMAX                                                  SAT17910
      MJ=1+(M-1)*KMAX2                                                  SAT17920
      TZMAX(I,M)=ABS(Z(I,MJ))                                           SAT17930
      DO 201 K=2,KMAX2                                                  SAT17940
      KM=K+(M-1)*KMAX2                                                  SAT17950
      AZTST=ABS(Z(I,KM))                                                SAT17960
      IF(AZTST.GT.TZMAX(I,M))  TZMAX(I,M)=AZTST                         SAT17970
  201 CONTINUE                                                          SAT17980
      NCONV=1                                                           SAT17990
      IF(ITRMAX.EQ.1) GO TO 66                                          SAT18000
      DO 1 M=1,MNMAXO                                                   SAT18010
      I=1+(KMAX+2)*(M-1)                                                SAT18020
      U1(M)=Z(1,I)                                                      SAT18030
      V1(M)=Z(2,I)                                                      SAT18040
      W1(M)=Z(3,I)                                                      SAT18050
      I1=I+1                                                            SAT18060
      U2(M)=Z(1,I1)                                                     SAT18070
      V2(M)=Z(2,I1)                                                     SAT18080
```

```
    1 W2(M)=Z(3,I1)                                                      SAT18090
      IF(IBCINL.LT.0) GO TO 100                                          SAT18100
      CALL PHIBET(1,Z,IS,JS,ID,JD,PHIXB,PHITB)                           SAT18110
      DO 2 M=1,MNMAX                                                     SAT18120
      BX1(M)=BX3(M)                                                      SAT18130
      BT1(M)=BT3(M)                                                      SAT18140
      BXT1(M)=BXT3(M)                                                    SAT18150
    2 BE1(M)=BE3(M)                                                      SAT18160
      CALL TEAETA(1,Z,IS,JS,ID,JD)                                       SAT18170
      DO 3 M=1,MNMAX                                                     SAT18180
      EXX1(M)=EXX3(M)                                                    SAT18190
      ETT1(M)=ETT3(M)                                                    SAT18200
      ETX1(M)=ETX3(M)                                                    SAT18210
      EXT1(M)=EXT3(M)                                                    SAT18220
      EX1(M)=EX3(M)                                                      SAT18230
    3 ET1(M)=ET3(M)                                                      SAT18240
  100  CALL PHIBET(2,Z,IS,JS,ID,JD,PHIXB,PHITB)                          SAT18250
      DO 4 M=1,MNMAX                                                     SAT18260
      BX2(M)=BX3(M)                                                      SAT18270
      BT2(M)=BT3(M)                                                      SAT18280
      BXT2(M)=BXT3(M)                                                    SAT18290
    4 BE2(M)=BE3(M)                                                      SAT18300
      CALL TEAETA(2,Z,IS,JS,ID,JD)                                       SAT18310
      DO 5 M=1,MNMAX                                                     SAT18320
      EXX2(M)=EXX3(M)                                                    SAT18330
      ETT2(M)=ETT3(M)                                                    SAT18340
      ETX2(M)=ETX3(M)                                                    SAT18350
      EXT2(M)=EXT3(M)                                                    SAT18360
      EX2(M)=EX3(M)                                                      SAT18370
    5 ET2(M)=ET3(M)                                                      SAT18380
      CALL PHIBET(3,Z,IS,JS,ID,JD,PHIXB,PHITB)                           SAT18390
      CALL TEAETA(3,Z,IS,JS,ID,JD)                                       SAT18400
   66 CONTINUE                                                           SAT18410
      IF(IBCINL.LT.0) GO TO 20                                           SAT18420
      CALL BDB(1,B1,DB,D,CD)                                             SAT18430
      GAM1=GAM(1)                                                        SAT18440
      CALL TLOAD(1,Z)                                                    SAT18450
      DO 8 M=1,MNMAX                                                     SAT18460
      IF(ITRMAX.EQ.1) GO TO 67                                           SAT18470
      FFS(1,M)=-TT(M)*ALOAD+OSE*(BX1(M)+BE1(M)+NU*(BT1(M)+BE1(M)))*B1    SAT18480
      FFS(2,M)=OSE*(B1   *D1      *BXT1(M)+EX1(M)+ET1(M))                SAT18490
      FFS(3,M)=LAM2*GAM1*D1*MT(M)*ALOAD-(EXX1(M)+ETX1(M))*SOE            SAT18500
      GO TO 8                                                            SAT18510
   67 FFS(1,M)=-TT(M)*ALOAD                                              SAT18520
      FFS(2,M)=0.                                                        SAT18530
      FFS(3,M)=LAM2*GAM1*D1*MT(M)*ALOAD                                  SAT18540
    8 FFS(4,M)=0.                                                        SAT18550
      DO 9 I=1,4                                                         SAT18560
```

```
    9 ELLS(I)=ALOAD*EL1(I)                                              SAT18570
      CALL FORCE(1,P,X,DEE,DST,Z,ZO,Z2,Z3)
   20 CALL FORCE(2,P,X,DEE,DST,Z,ZO,Z2,Z3)
      DO 10 K=3,KLL                                                     SAT18600
      KP=K+1                                                            SAT18610
      IF(ITRMAX.EQ.1) GO TO 10                                          SAT18620
      CALL UPDATE                                                       SAT18630
      CALL PHIBET(KP,Z,IS,JS,ID,JD,PHIXB,PHITB)                         SAT18640
      CALL TEAETA(KP,Z,IS,JS,ID,JD)                                     SAT18650
   10 CALL FORCE(K,P,X,DEE,DST,Z,ZO,Z2,Z3)
      IF(ITRMAX.NE.1) CALL UPDATE                                       SAT18670
      IF(IBCFNL.LT.0) GO TO 120                                         SAT18680
      IF(ITRMAX.EQ.1) GO TO 11                                          SAT18690
      CALL PHIBET(KMAX,Z,IS,JS,ID,JD,PHIXB,PHITB)                       SAT18700
      CALL TEAETA(KMAX,Z,IS,JS,ID,JD)                                   SAT18710
   11 CALL FORCE(KL,P,X,DEE,DST,Z,ZO,Z2,Z3)
      CALL FORCE(KMAX,P,X,DEE,DST,Z,ZO,Z2,Z3)
      DO 12 I=1,4                                                       SAT18740
   12 ELLS(I)=ALOAD*ELL(I)                                              SAT18750
      CALL BDB(KMAX,BL,DB,D,DD)                                         SAT18760
      GAML=GAM(KMAX)                                                    SAT18770
      FLS(4)=0.                                                         SAT18780
      CALL TLOAD(KMAX,Z)                                                SAT18790
      DO 14 M=1,MNMAX                                                   SAT18800
      IF(M.GT.1) ELLS(1)=0.0                                            SAT18810
      IF(ITRMAX.EQ.1) GO TO 68                                          SAT18820
      FLS(1)=-TT(M)*ALOAD+OSE*(BX3(M)+BE3(M)+NU*(ET3(M)+BE3(M)))*BL     SAT18830
      FLS(2)  =OSE*(BL*D1*BXT3(M)+EX3(M)+ET3(M))                        SAT18840
      FLS(3)=LAM2*GAML*D1*MT(M)*ALOAD-(EXX3(M)+ETX3(M))*SOE             SAT18850
      GO TO 69                                                          SAT18860
   68 FLS(1)=-TT(M)*ALOAD                                               SAT18870
      FLS(2)=0.                                                         SAT18880
      FLS(3)=LAM2*GAML*D1*MT(M)*ALOAD                                   SAT18890
   69 CONTINUE                                                          SAT18900
      IK=KL+KMAX*(M-1)                                                  SAT18910
      IJ=KMAX*M                                                         SAT18920
      L=M*KMAX2                                                         SAT18930
      DO 14 I=1,4                                                       SAT18940
      SUMZ=0.                                                           SAT18950
      DO 15 J=1,4                                                       SAT18960
C***********************************************************************SAT18970
C     THE FOLLOWING CARD CAUSES BOUNDARY CONS TO EXIST FOR MODE '0' ONLYSAT18980
C***********************************************************************SAT18990
      IF (M.NE.1) ELLS(J)=0                                             SAT19000
   15 SUMZ=SUMZ+ZF1M(I,J,M)*ELLS(J)+ZF2M(I,J,M)*X(J,IJ)+ZF3M(I,J,M)*    SAT19010
     1X(J,IK)+ZF4M(I,J,M)*FLS(J)                                        SAT19020
   14 Z(I,L)=SUMZ                                                       SAT19030
      LS=1                                                              SAT19040
```

```
150  DO 16 M=1,MNMAX                                                 SAT19050
     DO 16 L=LS,KMAX                                                 SAT19060
     K=KMAX2-L                                                       SAT19070
     KPX=K-1                                                         SAT19080
     KZ=K+1                                                          SAT19090
     IJ=KPX+(M-1)*KMAX                                               SAT19100
     JK=KZ+(M-1)*KMAX2                                               SAT19110
     KK=JK-1                                                         SAT19120
     DO 17 I=1,4                                                     SAT19130
     SUMZ=0.                                                         SAT19140
     DO 18 J=1,4                                                     SAT19150
  18 SUMZ=SUMZ-P(I,J,IJ)*Z(J,JK)                                     SAT19160
     SUMZ=SUMZ+X(I,IJ)                                               SAT19170
     ASUMZ=ABS(SUMZ)                                                 SAT19180
     IF(ASUMZ.GT.1.E+15) ITR=ITRMAX                                  SAT19190
     IF(NCONV.NE.1 .OR. ASUMZ .LT. 1.E-05) GO TO 17                  SAT19200
     DELZ=ABS(Z(I,KK)-SUMZ)                                          SAT19210
     ZTEST=EPS*TZMAX(I,M)                                            SAT19220
     IF(DELZ.GT.ZTEST) NCONV=0                                       SAT19230
  17 Z(I,KK)=SUMZ                                                    SAT19240
  16 CONTINUE                                                        SAT19250
     IF(IBCINL.LT.0) GO TO 30                                        SAT19260
     DO 25 M=1,MNMAX                                                 SAT19270
     CALL EFG(1,M,Z0,Z2,Z3)                                          SAT19280
     CALL ABC                                                        SAT19290
     IJ=2+(M-1)*KMAX2
     IJ1=IJ+1                                                        SAT19310
     IJ2=IJ-1                                                        SAT19320
     DO 21 I=1,4                                                     SAT19330
     SUMZ=0.                                                         SAT19340
     DO 22 J=1,4                                                     SAT19350
  22 SUMZ=SUMZ-A(I,J)*Z(J,IJ1)-BEE(I,J)*Z(J,IJ)                      SAT19360
  21 ZT(I)=SUMZ+GEES(I,M)                                            SAT19370
     CALL MATINV(C,4,ZT,1,DETERM,IPIVOT,INDEX,4,ISCALE)              SAT19380
     DO 23 I=1,4                                                     SAT19390
  23 Z(I,IJ2)=ZT(I)                                                  SAT19400
  25 CONTINUE                                                        SAT19410
  30 RETURN                                                          SAT19420
 100 CALL INLPOL (Z,PHIXB,PHITB)                                     SAT19430
     DO 101 M=1,MNMAXC                                               SAT19440
     U1(M)=U2(M)                                                     SAT19450
     V1(M)=V2(M)                                                     SAT19460
     W1(M)=W2(M)                                                     SAT19470
     IJ=3+KMAX2*(M-1)                                                SAT19480
     U2(M)=Z(1,IJ)                                                   SAT19490
     V2(M)=Z(2,IJ)                                                   SAT19500
 101 W2(M)=Z(3,IJ)                                                   SAT19510
     GO TO 102                                                       SAT19520
```

```
120 IF(ITRMAX.NE.1) CALL FNLPOL (Z,PHIXB,PHITB)                        SAT19530
    CALL FORCE(KL  ,P,X,DEE,DST,Z,ZC,Z2,Z3)
    IF(M2.EQ.0) GO TO 122                                              SAT19550
    L=KL+(M2-1)*KMAX                                                   SAT19560
    L1=KMAX1+(M2-1)*KMAX2                                              SAT19570
    DO 130 I=1,4                                                       SAT19580
    SUM=0.                                                             SAT19590
    DO 131 J=1,4                                                       SAT19600
131 SUM=SUM+CL2(I,J)*X(J,L)                                            SAT19610
    ASUMZ=ABS(SUM)                                                     SAT19620
    IF(NCONV.NE.1 .OR. ASUMZ.LT.1.E-05) GO TO 130                      SAT19630
    DELZ=ABS(Z(I,L1)-SUM)                                              SAT19640
    ZTEST=EPS*TZMAX(I,M2)                                              SAT19650
    IF(DELZ.GT.ZTEST) NCONV=0                                          SAT19660
130 Z(I,L1)=SUM                                                        SAT19670
122 IF(M1.EQ.0) GO TO 123                                              SAT19680
    L=KL+(M1-1)*KMAX                                                   SAT19690
    L1=KMAX1+(M1-1)*KMAX2                                              SAT19700
    DO 132 I=1,4                                                       SAT19710
    SUM=0.                                                             SAT19720
    DO 133 J=1,4                                                       SAT19730
133 SUM=SUM+CL1(I,J)*X(J,L)                                            SAT19740
    ASUMZ=ABS(SUM)                                                     SAT19750
    IF(NCONV.NE.1 .OR. ASUMZ .LT. 1.E-05) GO TO 132                    SAT19760
    DELZ=ABS(Z(I,L1)-SUM)                                              SAT19770
    ZTEST=EPS*TZMAX(I,M1)                                              SAT19780
    IF(DELZ.GT.ZTEST) NCONV=0                                          SAT19790
132 Z(I,L1)=SUM                                                        SAT19800
123 IF(M0.EQ.0) GO TO 124                                              SAT19810
    L=KL+(M0-1)*KMAX                                                   SAT19820
    L1=KMAX1+(M0-1)*KMAX2                                              SAT19830
    DO 134 I=1,4                                                       SAT19840
    SUM=0.                                                             SAT19850
    DO 135 J=1,4                                                       SAT19860
135 SUM=SUM+CL0(I,J)*X(J,L)                                            SAT19870
    ASUMZ=ABS(SUM)                                                     SAT19880
    IF(NCONV.NE.1 .OR. ASUMZ.LT.1.E-06) GO TO 134                      SAT19890
    DELZ=ABS(Z(I,L1)-SUM)                                              SAT19900
    ZTEST=EPS*TZMAX(I,M0)                                              SAT19910
    IF(DELZ.GT.ZTEST) NCONV=0                                          SAT19920
134 Z(I,L1)=SUM                                                        SAT19930
124 LS=2                                                               SAT19940
    GO TO 150                                                          SAT19950
    END                                                                SAT19960
                                                                       SAT19970
    SUBROUTINE PLOT2(NTH)                                              SAT19980
C*********************************************************************SAT19990
C   THIS SUBROUTINE CALLS PLOTTING ROUTINES FOR APPROPRIATE (USER     *SAT20000
```

```
C      SPECIFIED) OUTPUT QUANTITIES                                      *
C***********************************************************************
       IMPLICIT LOGICAL*1 ($)
       COMMON /IBL4/ KMAX,KL
       COMMON /BL19/ TH(36)
       COMMON /BLPLOT/  IRADII,IGAMMA,IOMEGS,IOMEGT,IDEOMS,IBSTIF,IDSTIF,
     1                  IBBSTF,IDDSTF,IPR,IPS,IPT,ITT,IMT,IDTT,IDMT,INS,
     2                  INTH,INSTH,IQS,IMS,IMTH,IMSTH,IU,IV,IW,IPHIS,
     3                  IPHIT,IPHI,$PLOTS,$MOCAL
       COMMON /BLPLT1/  XRADII(200),YGAMMA(200),YOMEGS(200),YOMEGT(200),
     1                  YDEOMS(200),YBSTIF(200),YDSTIF(200),YBBSTF(200),
     2                  YDDSTF(200),YPR(200),YPS(200),YPT(200),YTT(200),
     3                  YMT(200),YDTT(200),YDMT(200),YNS(200),YNTH(200),
     4                  YNSTH(200),YQS(200),YMS(200),YMTH(200),YMSTH(200),
     5                  YU(200),YV(200),YW(200),YPHIS(200),YPHIT(200),
     6                  YPHI(200),XSTATN(200)
C***********************************************************************
       NGKMAX=-KMAX
       IF ($MOCAL) GO TO 121
       IF (INS.EQ.0) GO TO 4
       WRITE (6,1000)
       IF (INS.GT.0) CALL PLOTIT (XSTATN,YNS,KMAX,0)
       IF (INS.LT.0) CALL PLOTIT (XSTATN,YNS,NGKMAX,0)
       WRITE (6,1001)TH(NTH)
     4 IF (INTH.EQ.0) GO TO 5
       WRITE (6,1000)
       IF (INTH.GT.0) CALL PLOTIT (XSTATN,YNTH,KMAX,0)
       IF (INTH.LT.0) CALL PLOTIT (XSTATN,YNTH,NGKMAX,0)
       WRITE (6,1002) TH(NTH)
     5 IF (INSTH.EQ.0) GO TO 6
       WRITE (6,1000)
       IF (INSTH.GT.0) CALL PLOTIT (XSTATN,YNSTH,KMAX,0)
       IF (INSTH.LT.0) CALL PLOTIT (XSTATN,YNSTH,NGKMAX,0)
       WRITE (6,1003) TH(NTH)
     6 IF (IQS.EQ.0) GO TO 7
       WRITE (6,1000)
       IF (IQS.GT.0) CALL PLOTIT (XSTATN,YQS,KMAX,0)
       IF (IQS.LT.0) CALL PLOTIT (XSTATN,YQS,NGKMAX,0)
       WRITE (6,1004) TH(NTH)
     7 IF (IMS.EQ.0) GO TO 8
       WRITE (6,1000)
       IF (IMS.GT.0) CALL PLOTIT (XSTATN,YMS,KMAX,0)
       IF (IMS.LT.0) CALL PLOTIT (XSTATN,YMS,NGKMAX,0)
       WRITE (6,1005) TH(NTH)
     8 IF (IMTH.EQ.0) GO TO 9
       WRITE (6,1000)
       IF (IMTH.GT.0) CALL PLOTIT (XSTATN,YMTH,KMAX,0)
       IF (IMTH.LT.0) CALL PLOTIT (XSTATN,YMTH,NGKMAX,0)
```

```
      WRITE (6,1006) TH(NTH)                                           SAT20490
    9 IF (IMSTH.EQ.0) GO TO 1211                                       SAT20500
      WRITE (6,1000)                                                   SAT20510
      IF (IMSTH.GT.0) CALL PLOTIT (XSTATN,YMSTH,KMAX,0)                SAT20520
      IF (IMSTH.LT.0) CALL PLOTIT (XSTATN,YMSTH,NGKMAX,0)              SAT20530
      WRITE (6,1007) TH(NTH)                                           SAT20540
 1211 IF (IU.EQ.0) GO TO 10                                            SAT20550
      WRITE (6,1000)                                                   SAT20560
      IF (IU.GT.0) CALL PLOTIT (XSTATN,YU,KMAX,0)                      SAT20570
      IF (IU.LT.0) CALL PLOTIT (XSTATN,YU,NGKMAX,0)                    SAT20580
      WRITE (6,1010) TH(NTH)                                           SAT20590
   10 IF (IV.EQ.0) GO TO 11                                            SAT20600
      WRITE (6,1000)                                                   SAT20610
      IF (IV.GT.0) CALL PLOTIT (XSTATN,YV,KMAX,0)                      SAT20620
      IF (IV.LT.0) CALL PLOTIT (XSTATN,YV,NGKMAX,0)                    SAT20630
      WRITE (6,1009) TH(NTH)                                           SAT20640
   11 IF (IW.EQ.0) GO TO 12                                            SAT20650
      WRITE (6,1000)                                                   SAT20660
      IF (IW.GT.0) CALL PLOTIT (XSTATN,YW,KMAX,0)                      SAT20670
      IF (IW.LT.0) CALL PLOTIT (XSTATN,YW,NGKMAX,0)                    SAT20680
      WRITE (6,1008) TH(NTH)                                           SAT20690
   12 IF (IPHIS.EQ.0) GO TO 13                                         SAT20700
      WRITE (6,1000)                                                   SAT20710
      IF (IPHIS.GT.0) CALL PLOTIT (XSTATN,YPHIS,KMAX,0)                SAT20720
      IF (IPHIS.LT.0) CALL PLOTIT (XSTATN,YPHIS,NGKMAX,0)              SAT20730
      WRITE (6,1011) TH(NTH)                                           SAT20740
   13 IF (IPHIT.EQ.0) GO TO 14                                         SAT20750
      WRITE (6,1000)                                                   SAT20760
      IF (IPHIT.GT.0) CALL PLOTIT (XSTATN,YPHIT,KMAX,0)                SAT20770
      IF (IPHIT.LT.0) CALL PLOTIT (XSTATN,YPHIT,NGKMAX,0)              SAT20780
      WRITE (6,1012) TH(NTH)                                           SAT20790
   14 IF (IPHI.EQ.0) GO TO 21                                          SAT20800
      WRITE (6,1000)                                                   SAT20810
      IF (IPHI .GT.0) CALL PLOTIT (XSTATN,YPHI ,KMAX,0)                SAT20820
      IF (IPHI .LT.0) CALL PLOTIT (XSTATN,YPHI ,NGKMAX,0)              SAT20830
      WRITE (6,1013) TH(NTH)                                           SAT20840
   21 RETURN                                                           SAT20850
  121 IF (INS.EQ.0) GO TO 15                                           SAT20860
      WRITE (6,1000)                                                   SAT20870
      IF (INS.GT.0) CALL PLOTIT (XSTATN,YNS,KMAX,0)                    SAT20880
      IF (INS.LT.0) CALL PLOTIT (XSTATN,YNS,NGKMAX,0)                  SAT20890
      WRITE (6,2001)                                                   SAT20900
   15 IF (INTH.EQ.0) GO TO 16                                          SAT20910
      WRITE (6,1000)                                                   SAT20920
      IF (INTH.GT.0) CALL PLOTIT (XSTATN,YNTH,KMAX,0)                  SAT20930
      IF (INTH.LT.0) CALL PLOTIT (XSTATN,YNTH,NGKMAX,0)                SAT20940
      WRITE (6,2002)                                                   SAT20950
   16 IF (INSTH.EQ.0) GO TO 17                                         SAT20960
```

```
      WRITE (6,1000)                                                    SAT20970
      IF (INSTH.GT.0) CALL PLOTIT (XSTATN,YNSTH,KMAX,0)                 SAT20980
      IF (INSTH.LT.0) CALL PLOTIT (XSTATN,YNSTH,NGKMAX,0)               SAT20990
      WRITE (6,2003)                                                    SAT21000
   17 IF (IQS.EQ.0) GO TO 18                                            SAT21010
      WRITE (6,1000)                                                    SAT21020
      IF (IQS.GT.0) CALL PLOTIT (XSTATN,YQS,KMAX,0)                     SAT21030
      IF (IQS.LT.0) CALL PLOTIT (XSTATN,YQS,NGKMAX,0)                   SAT21040
      WRITE (6,2004)                                                    SAT21050
   18 IF (IMS.EQ.0) GO TO 19                                            SAT21060
      WRITE (6,1000)                                                    SAT21070
      IF (IMS.GT.0) CALL PLOTIT (XSTATN,YMS,KMAX,0)                     SAT21080
      IF (IMS.LT.0) CALL PLOTIT (XSTATN,YMS,NGKMAX,0)                   SAT21090
      WRITE (6,2005)                                                    SAT21100
   19 IF (IMTH.EQ.0) GO TO 22                                           SAT21110
      WRITE (6,1000)                                                    SAT21120
      IF (IMTH.GT.0) CALL PLOTIT (XSTATN,YMTH,KMAX,0)                   SAT21130
      IF (IMTH.LT.0) CALL PLOTIT (XSTATN,YMTH,NGKMAX,0)                 SAT21140
      WRITE (6,2006)                                                    SAT21150
   22 IF (IMSTH.EQ.0) GO TO 231                                         SAT21160
      WRITE (6,1000)                                                    SAT21170
      IF (IMSTH.GT.0) CALL PLOTIT (XSTATN,YMSTH,KMAX,0)                 SAT21180
      IF (IMSTH.LT.0) CALL PLOTIT (XSTATN,YMSTH,NGKMAX,0)               SAT21190
      WRITE (6,2007)                                                    SAT21200
  231 IF (IU.EQ.0) GO TO 23                                             SAT21210
      WRITE (6,1000)                                                    SAT21220
      IF (IU.GT.0) CALL PLOTIT (XSTATN,YU,KMAX,0)                       SAT21230
      IF (IU.LT.0) CALL PLOTIT (XSTATN,YU,NGKMAX,0)                     SAT21240
      WRITE (6,2010)                                                    SAT21250
   23 IF (IV.EQ.0) GO TO 24                                             SAT21260
      WRITE (6,1000)                                                    SAT21270
      IF (IV.GT.0) CALL PLOTIT (XSTATN,YV,KMAX,0)                       SAT21280
      IF (IV.LT.0) CALL PLOTIT (XSTATN,YV,NGKMAX,0)                     SAT21290
      WRITE (6,2009)                                                    SAT21300
   24 IF (IW.EQ.0) GO TO 25                                             SAT21310
      WRITE (6,1000)                                                    SAT21320
      IF (IW.GT.0) CALL PLOTIT (XSTATN,YW,KMAX,0)                       SAT21330
      IF (IW.LT.0) CALL PLOTIT (XSTATN,YW,NGKMAX,0)                     SAT21340
      WRITE (6,2008)                                                    SAT21350
   25 IF (IPHIS.EQ.0) GO TO 26                                          SAT21360
      WRITE (6,1000)                                                    SAT21370
      IF (IPHIS.GT.0) CALL PLOTIT (XSTATN,YPHIS,KMAX,0)                 SAT21380
      IF (IPHIS.LT.0) CALL PLOTIT (XSTATN,YPHIS,NGKMAX,0)               SAT21390
      WRITE (6,2011)                                                    SAT21400
   26 IF (IPHIT.EQ.0) GO TO 27                                          SAT21410
      WRITE (6,1000)                                                    SAT21420
      IF (IPHIT.GT.0) CALL PLOTIT (XSTATN,YPHIT,KMAX,0)                 SAT21430
      IF (IPHIT.LT.0) CALL PLOTIT (XSTATN,YPHIT,NGKMAX,0)               SAT21440
```

```
      WRITE (6,2012)                                                    SAT21450
   27 IF (IPHI.EQ.0) GO TO 28                                          SAT21460
      WRITE (6,1000)                                                    SAT21470
      IF (IPHI .GT.0) CALL PLOTIT (XSTATN,YPHI  ,KMAX,0)                SAT21480
      IF (IPHI .LT.0) CALL PLOTIT (XSTATN,YPHI  ,NGKMAX,0)              SAT21490
      WRITE (6,2013)                                                    SAT21500
   28 RETURN                                                            SAT21510
C***********************************************************************SAT21520
 1000 FORMAT ('1')                                                      SAT21530
 1001 FORMAT ('0',T10,'"S"-MEMBRANE FORCE VS STATN,  MERIDIAN AT THETA =SAT21540
     1 ',F10.5,'  RADIANS')                                             SAT21550
 1002 FORMAT ('0',T10,'"THETA"-MEMBRANE FORCE VS STATN,  MERIDIAN AT THESAT21560
     1TA = ',F10.5,'  RADIANS')                                         SAT21570
 1003 FORMAT ('0',T10,'"S-THETA" MEMBRANE FORCE VS STATN,  MERIDIAN AT TSAT21580
     1HETA = ',F10.5,'  RADIANS')                                       SAT21590
 1004 FORMAT ('0',T10,'EFFECTIVE SHEAR VS STATN,  MERIDIAN AT THETA = ',SAT21600
     1F10.5,'  RADIANS')                                                SAT21610
 1005 FORMAT ('0',T10,'"S"-BENDING MOMENT VS STATN,  MERIDIAN AT THETA =SAT21620
     1 ',F10.5,' RADIANS')                                              SAT21630
 1006 FORMAT ('0',T10,'THETA-BENDING MOMENT VS STATN,  MERIDIAN AT THETASAT21640
     1 = ',F10.5,'  RADIANS')                                           SAT21650
 1007 FORMAT ('0',T10,'S-THETA BENDING MOMENT VS STATN,  MERIDIAN AT THESAT21660
     1TA = ',F10.5,'  RADIANS')                                         SAT21670
 1008 FORMAT ('0',T10,'NORMAL DEFLECTION VS STATN,  MERIDIAN AT THETA = SAT21680
     1',F10.5,'  RADIANS')                                              SAT21690
 1009 FORMAT ('0',T10,'CIRCUMFERENTIAL DEFLECTION VS STATN,  MERIDIAN ATSAT21700
     1 THETA = ',F10.5,'  RADIANS')                                     SAT21710
 1010 FORMAT ('0',T10,'MERIDIONAL DEFLECTION VS STATN,  MERIDIAN AT THETSAT21720
     1A = ',F10.5,'  RADIANS')                                          SAT21730
 1011 FORMAT ('0',T10,'MERIDIONAL ROTATION VS STATN,  MERIDIAN AT THETA SAT21740
     1= ',F10.5,'  RADIANS')                                            SAT21750
 1012 FORMAT ('0',T10,'CIRCUMFERENTIAL ROTATION VS STATN,  MERIDIAN AT TSAT21760
     1HETA = ',F10.5,'  RADIANS')                                       SAT21770
 1013 FORMAT ('0',T10,'MERIDIO-CIRCUMFERENTIAL ROTATION VS STATN,  MERIDSAT21780
     1IAN AT THETA = ',F10.5,'  RADIANS')                               SAT21790
 2001 FORMAT ('0',T10,'"S"-MEMBRANE FORCE VS STATION')                  SAT21800
 2002 FORMAT ('0',T10,'"THETA"-MEMBRANE FORCE VS STATION')              SAT21810
 2003 FORMAT ('0',T10,'"S-THETA" MEMBRANE FORCE VS STATION')            SAT21820
 2004 FORMAT ('0',T10,'EFFECTIVE SHEAR VS STATION')                     SAT21830
 2005 FORMAT ('0',T10,'"S"-BENDING MOMENT VS STATION')                  SAT21840
 2006 FORMAT ('0',T10,'THETA-BENDING MOMENT VS STATION')                SAT21850
 2007 FORMAT ('0',T10,'S-THETA BENDING MOMENT VS STATION')              SAT21860
 2008 FORMAT ('0',T10,'NORMAL DEFLECTION VS STATION')                   SAT21870
 2009 FORMAT ('0',T10,'CIRCUMFERENTIAL DEFLECTION VS STATION')          SAT21880
 2010 FORMAT ('0',T10,'MERIDIONAL DEFLECTION VS STATION')               SAT21890
 2011 FORMAT ('0',T10,'MERIDIONAL ROTATION VS STATION')                 SAT21900
 2012 FORMAT ('0',T10,'CIRCUMFERENTIAL ROTATION VS STATION')            SAT21910
 2013 FORMAT ('0',T10,'MERIDIO-CIRCUMFERENTIAL ROTATION VS STATION')    SAT21920
```

```
C*************************************************************************SAT21930
      END                                                                 SAT21940
      SUBROUTINE PLOT1(I)                                                 SAT21950
C*************************************************************************SAT21960
C     THIS SUBROUTINE CALLS PLOTTING ROUTINES FOR APPROPRIATE (USER      *SAT21970
C     SPECIFIED) INPUT QUANTITIES                                        *SAT21980
C*************************************************************************SAT21990
      IMPLICIT LOGICAL*1 ($)                                              SAT22000
      COMMON /IBL4/ KMAX,KL                                               SAT22010
      COMMON /BLPLOT/ IRADII,IGAMMA,IOMEGS,IOMEGT,IDEOMS,IBSTIF,IDSTIF,   SAT22020
     1                IBBSTF,IDDSTF,IPR,IPS,IPT,ITT,IMT,IDTT,IDMT,INS,    SAT22030
     2                INTH,INSTH,IQS,IMS,IMTH,IMSTH,IU,IV,IW,IPHIS,       SAT22040
     3                IPHIT,IPHI,$PLOTS,$MODAL                            SAT22050
      COMMON /BLPLT1/ XRADII(200),YGAMMA(200),YOMEGS(200),YOMEGT(200),    SAT22060
     1                YDEOMS(200),YBSTIF(200),YDSTIF(200),YBBSTF(200),    SAT22070
     2                YDDSTF(200),YPR(200),YPS(200),YPT(200),YTT(200),    SAT22080
     3                YMT(200),YDTT(200),YDMT(200),YNS(200),YNTH(200),    SAT22090
     4                YNSTH(200),YQS(200),YMS(200),YMTH(200),YMSTH(200),SAT22100
     5                YU(200),YV(200),YW(200),YPHIS(200),YPHIT(200),      SAT22110
     6                YPHI(200),XSTATN(200)                               SAT22120
C*************************************************************************SAT22130
      NGKMAX=-KMAX                                                        SAT22140
      IF (I.GT.1) GO TO 1                                                 SAT22150
      IF (IRADII.EQ.0) GO TO 2                                            SAT22160
      WRITE (6,1000)                                                      SAT22170
      CALL PLOTIT (XSTATN,XRADII,NGKMAX,0)                                SAT22180
      WRITE (6,1001)                                                      SAT22190
    2 IF (IGAMMA.EQ.0) GO TO 4                                            SAT22200
      WRITE (6,1000)                                                      SAT22210
      CALL PLOTIT (XSTATN,YGAMMA,NGKMAX,0)                                SAT22220
      WRITE (6,1002)                                                      SAT22230
    4 IF (IOMEGS.EQ.0) GO TO 5                                            SAT22240
      WRITE (6,1000)                                                      SAT22250
      CALL PLOTIT (XSTATN,YOMEGS,NGKMAX,0)                                SAT22260
      WRITE (6,1003)                                                      SAT22270
    5 IF (IOMEGT.EQ.0) GO TO 6                                            SAT22280
      WRITE (6,1000)                                                      SAT22290
      CALL PLOTIT (XSTATN,YOMEGT,NGKMAX,0)                                SAT22300
      WRITE (6,1004)                                                      SAT22310
    6 IF (IDEOMS.EQ.0) GO TO 7                                            SAT22320
      WRITE (6,1000)                                                      SAT22330
      CALL PLOTIT (XSTATN,YDEOMS,NGKMAX,0)                                SAT22340
      WRITE (6,1005)                                                      SAT22350
    7 IF (IBSTIF.EQ.0) GO TO 8                                            SAT22360
      WRITE (6,1000)                                                      SAT22370
      CALL PLOTIT (XSTATN,YBSTIF,NGKMAX,0)                                SAT22380
      WRITE (6,1006)                                                      SAT22390
    8 IF (IDSTIF.EQ.0) GO TO 9                                            SAT22400
```

```
      WRITE (6,1000)                                                    SAT22410
      CALL PLCTIT (XSTATN,YDSTIF,NGKMAX,0)                              SAT22420
      WRITE (6,1007)                                                    SAT22430
    9 IF (IBBSTF.EQ.0) GC TO 10                                         SAT22440
      WRITE (6,1000)                                                    SAT22450
      CALL PLCTIT (XSTATN,YBBSTF,NGKMAX,0)                              SAT22460
      WRITE (6,1008)                                                    SAT22470
   1C IF (IDCSTF.EQ.0) GC TO 1                                          SAT22480
      WRITE (6,1C00)                                                    SAT22490
      CALL PLCTIT (XSTATN,YDCSTF,NGKMAX,0)                              SAT22500
      WRITE (6,1C09)                                                    SAT22510
    1 IF (IPR.EQ.0) GO TC 11                                            SAT22520
      WRITE (6,1000)                                                    SAT22530
      CALL PLCTIT (XSTATN,YPR,NGKMAX,0)                                 SAT22540
      WRITE (6,1010)                                                    SAT22550
   11 IF (IPS.EQ.0) GO TC 12                                            SAT22560
      WRITE (6,1CC0)                                                    SAT22570
      CALL PLCTIT (XSTATN,YPS,NGKMAX,0)                                 SAT22580
      WRITE (6,1011)                                                    SAT22590
   12 IF (IPT.EQ.0) GO TO 13                                            SAT22600
      WRITE (6,1000)                                                    SAT22610
      CALL PLCTIT (XSTATN,YPT,NGKMAX,0)                                 SAT22620
      WRITE (6,1012)                                                    SAT22630
   13 IF (ITT.EQ.0) GO TC 14                                            SAT22640
      WRITE (6,1000)                                                    SAT22650
      CALL PLCTIT (XSTATN,YTT,NGKMAX,0)                                 SAT22660
      WRITE (6,1013)                                                    SAT22670
   14 IF (IMT.EQ.0) GO TO 15                                            SAT22680
      WRITE (6,1000)                                                    SAT22690
      CALL PLCTIT (XSTATN,YMT,NGKMAX,0)                                 SAT22700
      WRITE (6,1014)                                                    SAT22710
   15 IF (IDTT.EQ.0) GO TC 16                                           SAT22720
      WRITE (6,1000)                                                    SAT22730
      CALL PLCTIT (XSTATN,YDTT,NGKMAX,0)                                SAT22740
      WRITE (6,1015)                                                    SAT22750
   16 IF (ICMT.EQ.0) GO TO 17                                           SAT22760
      WRITE (6,1000)                                                    SAT22770
      CALL PLCTIT (XSTATN,YDMT,NGKMAX,0)                                SAT22780
      WRITE (6,1016)                                                    SAT22790
      WRITE (6,1000)                                                    SAT22800
   17 RETURN                                                            SAT22810
C***********************************************************************SAT22820
 1000 FORMAT ('1')                                                      SAT22830
 1001 FORMAT ('0',T10,'RACIUS VS STATICN')                              SAT22840
 1002 FORMAT ('0',T10,'GAMMA VS STATICN')                               SAT22850
 1003 FORMAT ('0',T10,'OMEGA-S VS STATICN')                             SAT22860
 1004 FORMAT ('0',T10,'OMEGA-THETA VS STATION')                         SAT22870
 1005 FORMAT ('0',T10,'DECMEGA-S VS STATION')                           SAT22880
```

```
1006 FORMAT ('0',T10,'B-STIFFNESS VS STATION')                        SAT22890
1007 FORMAT ('0',T10,'D-STIFFNESS VS STATION')                        SAT22900
1008 FORMAT ('0',T10,'DB-STIFFNESS VS STATION')                       SAT22910
1009 FORMAT ('0',T10,'DD-STIFFNESS VS STATION')                       SAT22920
1010 FORMAT ('0',T10,'NORMAL LOADING VS STATION')                     SAT22930
1011 FORMAT ('0',T10,'MERIDIO-TANGENTIAL LOADING VS STATION')         SAT22940
1012 FORMAT ('0',T10,'CIRCUMFERENTIO-TANGENTIAL LOADING VS STATION')  SAT22950
1013 FORMAT ('0',T10,'THERMAL LOADING VS STATION')                    SAT22960
1014 FORMAT ('0',T10,'THERMAL BENDING LOADING VS STATION')            SAT22970
1015 FORMAT ('0',T10,'DE-THERMAL LOADING VS STATION')                 SAT22980
1016 FORMAT ('0',T10,'DE-THERMAL-BENDING LOADING VS STATION')         SAT22990
C***********************************************************************SAT23000
      END                                                             SAT23010
      SUBROUTINE FLYNVY                                               00000010
      WRITE (6,111)                                                   00000020
      WRITE (6,1)                                                     00000030
      WRITE (6,2)                                                     00000040
      WRITE (6,3)                                                     00000050
      WRITE (6,4)                                                     00000060
      WRITE (6,5)                                                     00000070
      WRITE (6,6)                                                     00000080
      WRITE (6,7)                                                     00000090
      WRITE (6,8)                                                     00000100
      WRITE (6,9)                                                     00000110
      WRITE (6,10)                                                    00000120
      WRITE (6,11)                                                    00000130
      WRITE (6,12)                                                    00000140
      WRITE (6,13)                                                    00000150
      WRITE (6,14)                                                    00000160
      WRITE (6,15)                                                    00000170
      WRITE (6,16)                                                    00000180
      WRITE (6,17)                                                    00000190
      WRITE (6,18)                                                    00000200
      WRITE (6,19)                                                    00000210
      WRITE (6,20)                                                    00000220
      WRITE (6,21)                                                    00000230
      WRITE (6,22)                                                    00000240
      WRITE (6,23)                                                    00000250
      WRITE (6,24)                                                    00000260
      WRITE (6,25)                                                    00000270
      WRITE (6,26)                                                    00000280
      WRITE (6,27)                                                    00000290
      WRITE (6,28)                                                    00000300
      WRITE (6,29)                                                    00000310
      WRITE (6,30)                                                    00000320
      WRITE (6,31)                                                    00000330
      WRITE (6,32)                                                    00000340
      WRITE (6,33)                                                    00000350
```

```
      WRITE (6,34)
      WRITE (6,35)
      WRITE (6,36)
      WRITE (6,37)
      WRITE (6,38)
      WRITE (6,39)
      WRITE (6,40)
  111 FORMAT ('1')
    1 FORMAT (' ',T11,'
     1       XXX')
    2 FORMAT (' ',T11,'
     1      XXXXX')
    3 FORMAT (' ',T11,'
     1    XXX   XXX')
    4 FORMAT (' ',T11,'
     1    XXX   XXX')
    5 FORMAT (' ',T11,'
     1      X@XXX')
    6 FORMAT (' ',T11,'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
     1XXXXXXX@XXXXXXXXXXXX    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
     2XX')
    7 FORMAT (' ',T11,'   XXXXXXXXCOXXXXXXXOOXXXXXXXXOCXXXXXXXXXXCX
     1XXXXXXX@XXXXXXXXXXXX     XOXXXXXXXXXXOOXXXXXXXXOOXXXXXXXOCXXXXXXXX'
     2)
    8 FORMAT (' ',T11,'       XXXXXXOOXXXXXXXOOXXXXXXXXOCXXXXXXCX
     1       @XXX               XOXXXXXXCOXXXXXXXXOCXXXXXXXXOOXXXXXX')
    9 FORMAT (' ',T11,'           XXXXCCXXXXXXXCOXXXXXXXXCCXXXCXX
     1     XXXXXXXX             XXOXXXCCXXXXXXXXOCXXXXXXXCCXXXX')
   10 FORMAT (' ',T11,'               XXXXXXXXXXXXOCXXXXXXXXCOXXXXXXXXXXXX
     1XXXXXXOCCOOOOXXXXXXXXXXXXXXXXXXXOOXXXXXXXOCXXXXXXXXXXXX')
   11 FORMAT (' ',T11,'                   XXXXXXXXXXXOOXXXXXXOOXXXXXXXOO
     1CCCCCCOCCOOCCOOCOCCCOXXXXXXXXCOXXXXXXXCOXXXXXXXXXXXX')
   12 FORMAT (' ',T11,'                        XXXXXXXXXOCXXXXXOCXXXXXXOC
     1CCCCOCOCCCCCCCOCCCCCOXXXXXOCXXXXXXCCXXXXXXXXXX')
   13 FORMAT (' ',T11,'                             XXXXXXXCCXXXXOCXXXXO
     1CCCCCCCCCCCCCOOCCOXXXXOOXXXXOCXXXXXXX')
   14 FORMAT (' ',T11,'                                   XXXXOCXXXCOXXX
     1CX X X X X X X X XCCXXOCXXXCOXXXX')
   15 FORMAT (' ',T11,'                                        XXOXXXXOOX
     1CC X X X X X X X CCXOOXXXXOXX')
   16 FORMAT (' ',T11,'                                        CCCCCXXXXX
     1CC X X X X X X X CCXXXXXCOOOO')
   17 FORMAT (' ',T11,'                                         CCC
     1XCCX X X X X X XOCX      OOC')
   18 FORMAT (' ',T11,'                                          CCC
     1 XCC X X X X X OOX      OOO')
   19 FORMAT (' ',T11,'                                           OOC
     1  XCOX X X X XOOX      OOO')
```

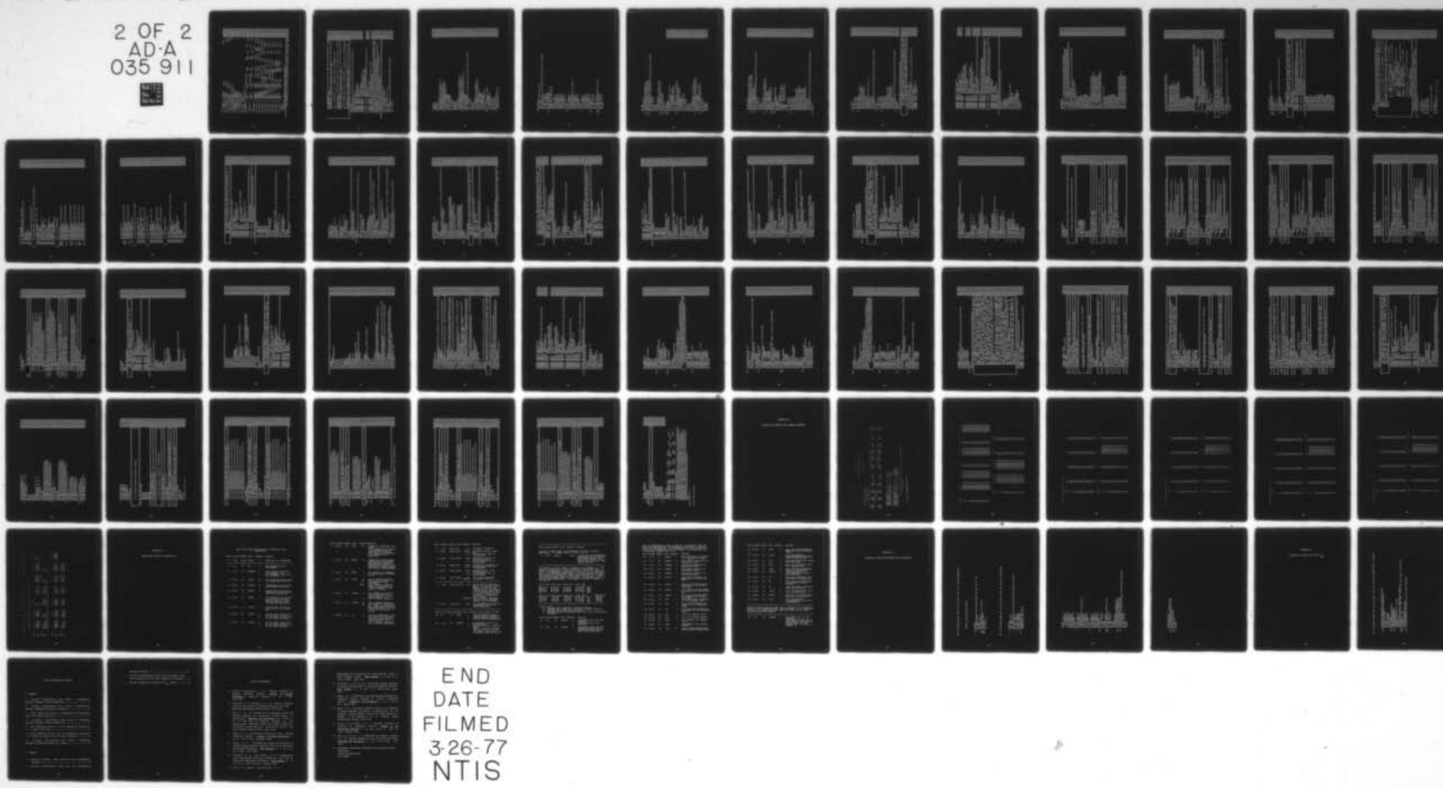
AD-A035 911 NAVAL POSTGRADUATE SCHOOL MONTEREY CALIF F/G 20/11

STATIC AND DYNAMIC BUCKLING OF SHALLOW SPHERICAL SHELLS SUBJECT--ETC(U)

DEC 76 M D SHUTT

UNCLASSIFIED NL

2 OF 2
AD-A
035 911



END
DATE
FILMED
3-26-77
NTIS

```
20 FORMAT (' ',T11,'                                                      OCO
 1    XOO X X X OOX       OOO')
21 FORMAT (' ',T11,'                                                       CCCO
 1      XOC X OCX       OOOC')
22 FORMAT (' ',T11,'                                                        CCCC
 1       XCOOCOX       OOOO')
23 FORMAT (' ',T11,'                                                          OO
 1CCC   COCXCXOOO  COOOO')
24 FORMAT (' ',T11,'
 1 CCCCOCCCOCCOOOO')
25 FORMAT (' ',T11,'
 1    COCCCCCCOOOO')
26 FORMAT (' ',T11,'
 1      COCCCCOO')
27 FORMAT (' ',T11,'
 1          OOO')
28 FORMAT (' ',//////////)
29 FORMAT (' ',T11,'      FFFFFFFFFFF  LLL         YYY          YYY
 1          NNN        NNN        A    VVV         VVV   YYY           YYY
 2')
30 FORMAT (' ',T11,'      FFFFFFFFFFF  LLL          YYY         YYY
 1          NNNN       NNN       AAA   VVV          VVV    YYY         YYY'
 2)
31 FORMAT (' ',T11,'      FFF          LLL           YYY      YYY
 1          NNNNN      NNN      AAAAA    VVV      VVV      YYY       YYY')
32 FORMAT (' ',T11,'      FFF          LLL            YYY    YYY
 1          NNNNNN     NNN      AAAAA    VVV      VVV       YYY    YYY')
33 FORMAT (' ',T11,'      FFF          LLL             YYY YYY
 1          NNNNNNN    NNN     AAA AAA    VVV    VVV         YYY YYY')
34 FORMAT (' ',T11,'      FFFFFFFF     LLL              YYYYY
 1          NNN NNNN   NNN     AAA AAA    VVV    VVV          YYYYY')
35 FORMAT (' ',T11,'      FFFFFFFF     LLL               YYY
 1          NNN  NNNN NNN    AAA   AAA    VVV VVV            YYY')
36 FORMAT (' ',T11,'      FFF          LLL               YYY
 1          NNN   NNNNNNN    AAAAAAAAA    VVV VVV            YYY')
37 FORMAT (' ',T11,'      FFF          LLL               YYY
 1          NNN    NNNNNN   AAAAAAAAAAA    VVVVV             YYY')
38 FORMAT (' ',T11,'      FFF          LLL               YYY
 1          NNN     NNNNN   AAA     AAA    VVVVV             YYY')
39 FORMAT (' ',T11,'      FFF          LLLLLLLLLLL       YYY
 1          NNN      NNNN  AAA      AAA     VVV              YYY')
40 FORMAT (' ',T11,'      FFF          LLLLLLLLLLL       YYY
 1          NNN       NNN  AAA      AAA      V               YYY')
   RETURN
   END
   SUBROUTINE PMATRX (P,X,ZO,Z2,Z3,CEE,DST)
C***********************************************************************
C       THIS SUBROUTINE CALLS THE SUBROUTINES FJ(K,MN), EFG(K,MN), ABC,
```

```
C         AND PANDD(K,MN) TO SET UP THE P, P-BAR AND P-HAT MATRICES GIVEN
C         BY EQUATIONS (30).                                            *
C         INTERNALLY, MATRICES DL, DG AND DF ARE SET UP FOR THE CALCULA-*
C         TION OF X(1) GIVEN BY EQUATION (31A), WHERE                   *
C                                                                       *
C         X(1) = DL*SMALL-L(1) + DG*SMALL-G(1) + DF*SMALL-F(1)          *
C                                                                       *
C         THE SPECIAL P MATRIX FOR A SHELL WITH AN INITIAL POLE IS ALSO *
C         COMPLETED HERE.                                               *
C         MATRICES ZF1M,ZF2M,ZF3M,ZF4M ARE SET UP FOR THE CALCULATION OF*
C         Z(K+1) GIVEN BY EQUATION (31B), WHERE                         *
C                                                                       *
C         Z(K+1)=ZF1M*SMALL-L(K) + ZF2M*X(K) + ZF3M*X(K-1) + ZF4M*SMALL-F
C                                                                       *
C         IF THE SHELL HAS A FINAL POLE, THE MATRICES CL0,CL1,CL2 ARE   *
C         PREPARED FOR THE CALCULATION OF Z(K)                          *
C************************************************************************
      REAL JAY
      DIMENSION P(4,4,1),DEE(4,4,1),DST(4,4,1),X(4,1),ZC(4,1),
     1Z2(4,1),Z3(4,1)
      COMMON /IBL1/ MNMAX
      COMMON /IBL2/ N(99),MNINIT
      COMMON /IBL3/ M0,M1,M2,M3
      COMMON /IBL4/ KMAX,KL
      COMMON /IBL5/ IBCINL,IBCFNL
      COMMON /BL1/  A(4,4),BEE(4,4),C(4,4)
      COMMON /BL4/  ZF1M(4,4,99),ZF2M(4,4,99),
     1              ZF3M(4,4,99),ZF4M(4,4,99)
      COMMON /BL13/ OMEG1(4,4),CAPL1(4,4),OMEGL(4,4),CAPLL(4,4),
     1              UNIT(4,4)
      COMMON /BL23/ JAY(4,4),H(4,4)
      COMMON /BL24/ DL(4,4,99),DG(4,4,99),DF(4,4,99)
      COMMON /BL25/ E(4,4),F(4,4),G(4,4)
      DIMENSION PATA(4,4),PBTA(4,4),POTA(4,4),PJTA(4,4),DLL(4,4),PTR(4,
     14),CGG(4,4),ZF1(4,4),ZF2(4,4),ZFP0(4,4),ZFP1(4,4),ZFP2(4,4),IPIVO
     2T(4),INDEX(4,2),CL0(4,4),CL1(4,4),CL2(4,4),G1(4)
      EQUIVALENCE (CL0(1),ZF1M(1)),(CL1(1),ZF2M(1)),(CL2(1),ZF3M(1)),
     1(ZFP0(1),PATA(1)),(ZFP1(1),PBTA(1)),(ZFP2(1),POTA(1))
     2,(ZF1(1),DLL(1)),(ZF2(1),PTR(1))
C************************************************************************
      IF(IBCINL.LT.0) GO TO 10
      DO 1 MN=MNINIT,MNMAX
      CALL HJ(1,MN)
      CALL EFG(1,MN,Z0,Z2,Z3)
      CALL ABC
      CALL MATINV(C,4,G1,0,DETERM,IPIVOT,INDEX,4,ISCALE)
      DO 3 I=1,4
      DO 3 J=1,4
```

```
      SUMA=0.
      SUMB=0.
      SUMC=0.
      SUMJ=0.
      DO 4 L=1,4
      SUMJ=SUMJ+OMEG1(I,L)*JAY(L,J)
      SUMA=SUMA+C(I,L)*A(L,J)
      SUMB=SUMB+C(I,L)*BEE(L,J)
    4 SUMC=SUMC+CMEG1(I,L)*H(L,J)
      PATA(I,J)=SUMA+UNIT(I,J)
      PBTA(I,J)=SUMB
      PCTA(I,J)=SUMC
    3 PJTA(I,J)=SUMJ
      DO 5 I=1,4
      DO 5 J=1,4
      SUMOB=0.
      SUMOA=0.
      SUMOC=0.
      DO 6 L=1,4
      SUMOB=SUMOB+POTA(I,L)*PBTA(L,J)
      SUMOA=SUMOA+POTA(I,L)*PATA(L,J)
    6 SUMOC=SUMOC+POTA(I,L)*C(L,J)
      DLL(I,J)=SUMOB+PJTA(I,J)+CAPL1(I,J)
      PTR(I,J)=SUMOA
    5 DGG(I,J)=SUMOC
      CALL MATINV(DLL,4,PTR,4,DETERM,IPIVOT,INDEX,4,ISCALE)
      DO 1 I=1,4
      DO 1 J=1,4
      SUMD=0.
      SUME=0.
      DO 7 L=1,4
      SUMD=SUMD+DLL(I,L)*DGG(L,J)
    7 SUME=SUME-DLL(I,L)*CMEG1(L,J)
      DL(I,J,MN)=DLL(I,J)
      DG(I,J,MN)=SUMD
      DF(I,J,MN)=SUME
      IJ=1+KMAX*(MN-1)
    1 P(I,J,IJ)=PTR(I,J)
      GO TO 20
   10 DO 11 MN=MNINIT,MNMAX
      NN=IABS(N(MN))
      IJ=1+KMAX*(MN-1)
      DO 14 I=1,4
      X(I,IJ)=0.
      DO 14 J=1,4
   14 P(I,J,IJ)=0.
      IF(NN.GT.3) GO TO 11
      IF(NN.GT.2) GO TO 90
```

```
C     IN FMATRX
      CALL EFG(2,MN,ZG,Z2,Z3)
      CALL ABC
      CALL MATINV(A,4,G1,0,DETERM,IPIVCT,INDEX,4,ISCALE)
      DO 901 II=1,4
      DO 901 JJ=1,4
      DL(II,JJ,MN)=0.
      DG(II,JJ,MN)=0.
  901 DF(II,JJ,MN)=0.
      IF(NN.GT.1) GO TO 12
      IF(NN.GT.0) GO TO 13
      M0=MN
      DL(1,1,MN)=1.
      DL(2,2,MN)=1.
      DL(3,3,MN)=-3.
      DL(4,4,MN)=-3.
      DG(3,3,MN)=4.
      DG(4,4,MN)=4.
      DF(3,3,MN)=-1.
      DF(4,4,MN)=-1.
      GO TO 902
   13 M1=MN
      DL(1,1,MN)=-3.
      DL(2,1,MN)=1.
      DL(2,2,MN)=1.
      IF(N(M1).LT.0)DL(2,2,MN)=-1.
      DL(3,3,MN)=1.
      DL(4,4,MN)=1.
      DG(1,1,MN)=4.
      DF(1,1,MN)=-1.
      GO TO 902
   12 M2=MN
      DL(1,1,MN)=1.
      DL(2,2,MN)=1.
      DL(3,3,MN)=1.
      DL(4,4,MN)=-3.
      DG(4,4,MN)=4.
      DF(4,4,MN)=-1.
  902 CONTINUE
      DO 903 II=1,4
      DO 903 JJ=1,4
      TTP=0.
      DO 904 L=1,4
  904 TTP=TTP+DF(II,L,MN)*A(L,JJ)
  903 CLO(II,JJ)=TTP
      DO 905 II=1,4
      DO 905 JJ=1,4
      TTP=0.
```

```
      TTQ=0.
      DO 906 L=1,4
      TTP=TTP+CLO(II,L)*C(L,JJ)
  906 TTQ=TTQ+CLO(II,L)*BEE(L,JJ)
      CL1(II,JJ)=DL(II,JJ,MN)-TTP
  905 CL2(II,JJ)=DG(II,JJ,MN)-TTQ
      CALL MATINV(CL1,4,G1,0,DETERM,IPIVOT,INDEX,4,ISCALE)
      DO 907 II=1,4
      DO 907 JJ=1,4
      TTP=0.
      TTQ=0.
      DO 908 L=1,4
      TTP=TTP+CL1(II,L)*CLO(L,JJ)
  908 TTQ=TTQ+CL1(II,L)*CL2(L,JJ)
      DL(II,JJ,MN)=-TTP
  907 P(II,JJ,IJ)=TTQ
      GO TO 11
   90 M3=MN
   11 CONTINUE
   20 KLAST=KMAX
      IF(IBCFNL.LT.0) KLAST=KL
      DO 23 K=2,KLAST
      DO 23 MN=MNINIT,MNMAX
      CALL EFG(K,MN,Z0,Z2,Z3)
      CALL ABC
   23 CALL PANDD(K,MN,P,CEE,DST,X)
      IF(IBCFNL.LT.0) GO TO 30
      DO 40 MN=MNINIT,MNMAX
      IKL=MN*KMAX-1
      JKL=KMAX*MN
      CALL FJ(KMAX,MN)
      DO 41 I=1,4
      DO 41 J=1,4
      SUMO=0.
      SUMP=0.
      SUMJ=0.
      DO 42 L=1,4
      SUMO=SUMO+CMEGL(I,L)*H(L,J)
      SUMP=SUMP+P(I,L,IKL)*F(L,J,JKL)
   42 SUMJ=SUMJ+CMEGL(I,L)*JAY(L,J)
      PATA(I,J)=SUMO
      PETA(I,J)=UNIT(I,J)-SUMP
   41 PJTA(I,J)=SUMJ+CAPLL(I,J)
      DO 43 I=1,4
      DO 43 J=1,4
      SUMOP=0.
      SUMJP=0.
      SUMOM=0.
```

```
      CC 44 L=1,4                                                  0000273C
      SUMOP=SLMOP+PATA(I,L)*PBTA(L,J)                              00002740
      SUMJP=SUMJP+PJTA(I,L)*P(L,J,JKL)                             00002750
   44 SUMCM=SLMOM-PATA(I,L)*P(L,J,IKL)                             00002760
      ZF1(I,J)=SUMOP-SUMJP                                         00002770
   43 ZF2(I,J)=SUMOM-PJTA(I,J)                                     0000278C
      CALL MATINV(ZF1,4,ZF2,4,DETERM,IFIVOT,INCEX,4,ISCALE)        00002790
      CC 45 I=1,4                                                  00002800
      CC 45 J=1,4                                                  00002810
      SZF3=0.                                                      00002820
      SZF4=0.                                                      00002830
      CC 46 L=1,4                                                  00002840
      SZF3=SZF3+ZF1(I,L)*FATA(L,J)                                 00002850
   46 SZF4=SZF4-ZF1(I,L)*CMEGL(L,J)                                00C02860
      ZF3M(I,J,MN)=SZF3                                            00002870
      ZF4M(I,J,MN)=SZF4                                            00002880
      ZF1M(I,J,MN)=ZF1(I,J)                                        0000289C
   45 ZF2M(I,J,MN)=ZF2(I,J)                                        00002900
   4C CCNTINUE                                                     00002910
      RETURN                                                       0C002920
   3C DC 31 MN=MNINIT,MNMAX                                        00002930
      IKL=MN*KMAX-1                                                00002940
      NN=N(MN)                                                     00002950
      NN=IABS(N(MN))                                               00002960
      IF(NN    .GT.3) GO TO 31                                     00002970
      IF(NN.GT.2) GO TO 300                                        00002980
      IF(NN    .GT.1) GO TO 33                                     00002990
      IF(NN    .GT.0) GC TC 34                                     00003000
      MC=MN                                                        00003010
      CC 35 J=1,4                                                  0C003020
      CC 35 I=1,4                                                  0C003030
      CLC(I,J)=0.                                                  00003040
   35 ZFPO(I,J)=0.                                                 00003050
      ZFFC(1,I)=1.                                                 00003060
      ZFPO(2,2)=1.                                                 00003070
      ZFFC(3,1)=P(3,1,IKL)                                         00003080
      ZFPC(3,2)=P(3,2,IKL)                                         00003090
      ZFPO(3,3)=P(3,3,IKL)+1.                                      00003100
      ZFFO(3,4)=P(3,4,IKL)                                         00003110
      ZFFC(4,1)=P(4,1,IKL)                                         00003120
      ZFFO(4,2)=P(4,2,IKL)                                         00003130
      ZFPO(4,3)=P(4,3,IKL)                                         00C03140
      ZFPO(4,4)=P(4,4,IKL)+1.                                      00003150
      CLC(3,3)=1.                                                  00003160
      CLC(4,4)=1.                                                  00003170
      CALL MATINV(ZFPO,4,CLO,4,DETERM,IFIVOT,INDEX,4,ISCALE)       00003180
      GC TO 31                                                     00003190
  3CC M3=MN                                                        00003200
```

```
      GO TO 31
   34 M1=MN
      DO 60 J=1,4
      DO 60 I=1,4
      CL1(I,J)=0.
   60 ZFP1(I,J)=0.
      ZFP1(1,1)=P(1,1,IKL)+1.
      ZFP1(1,2)=P(1,2,IKL)
      ZFP1(1,3)=P(1,3,IKL)
      ZFP1(1,4)=P(1,4,IKL)
      ZFP1(2,1)=1.
      ZFP1(2,2)=-1.
      IF(N(MN).LT.0) ZFP1(2,2) = 1.
      ZFP1(3,3)=1.
      ZFP1(4,4)=1.
      CL1(1,1)=1.
      CALL MATINV(ZFP1,4,CL1,4,DETERM,IPIVOT,INDEX,4,ISCALE)
      GO TO 31
   33 M2=MN
      DO 70 J=1,4
      DO 70 I=1,4
      CL2(I,J)=0.
   70 ZFP2(I,J)=0.
      ZFP2(1,1)=1.
      ZFP2(2,2)=1.
      ZFP2(3,3)=1.
      ZFP2(4,1)=P(4,1,IKL)
      ZFP2(4,2)=P(4,2,IKL)
      ZFP2(4,3)=P(4,3,IKL)
      ZFP2(4,4)=P(4,4,IKL)+1.
      CL2(4,4)=1.
      CALL MATINV(ZFP2,4,CL2,4,DETERM,IPIVOT,INDEX,4,ISCALE)
   31 CONTINUE
      RETURN
      END
      SUBROUTINE FORCE (K,F,X,DEE,DST,Z,ZC,Z2,Z3)
C**********************************************************************
C         THIS SUBROUTINE COMPUTES THE GEE VECTOR IN EQUATION (28), AND *
C         THE X VECTOR IN EQUATION (29A) FOR A GIVEN MERIDIONAL STATION *
C         K.  THE VECTOR GEES IS THE NON-LINEAR VALUE OF GEE AT STA. 1. *
C**********************************************************************
      IMPLICIT LOGICAL*1 ($)
      REAL NU,MT,LAM2,MASS,MAS
      DIMENSION P(4,4,1),DEE(4,4,1),DST(4,4,1),Z(4,1),ZC(4,1),
     1Z2(4,1),Z3(4,1),X(4,1)
      COMMON /IBL1/ MNMAX
      COMMON /IBL2/ N(99),MNINIT
      COMMON /IBL4/ KMAX,KL
```

```
      COMMON /IBL5/ IBCINL,IBCFNL
      COMMON /IBL8/ LSTEP,ITR
      COMMON /IBL12/ KMAX1,KMAX2,NCONV
      COMMON /IBL13/ ITRMAX,LSMAX
      COMMON /BL3/  PR(99),PX(99),PT(99)
      COMMON /BL4/  ZF1M(4,4,99),ZF2M(4,4,99),
     1              ZF3M(4,4,99),ZF4M(4,4,99)
      COMMON /BL5/  TT(99),MT(99),DT(99),DMT(99)
      COMMON /BL6/  SOE,CSE,ALOAD
      COMMON /BL7/  D1,S1
      COMMON /BL8/  R(500),GAM(500),OMT(500)
      COMMON /BL9/  FFS(4,99),EL1S(4),GEES(4,99)
      COMMON /BL11/ OMXI(500),PHEE,TO,T2
      COMMON /BL12/ TDLI,TDEL
      COMMON /BL14/ LAM2,LSD18,LSD1N
      COMMON /BL15/ NU,U1(99),V1(99),W1(99),V2(99),U2(99),W2(99),U3(99),
     1              V3(99),W3(99)
      COMMON /BL17/ DEL
      COMMON /BL24/ DL(4,4,99),DG(4,4,99),DF(4,4,99)
      COMMON /BL27/ BX3(99),BT3(99),BXT3(99),BE3(99)
      COMMON /BL28/ EXX3(99),ETT3(99),ETX3(99),EXT3(99),EX3(99),ET3(99)
      COMMON /BL29/ BX1(99),BT1(99),BXT1(99),BE1(99),BX2(99),BT2(99),
     1              BXT2(99),BE2(99)
      COMMON /BL30/ EXX1(99),ETT1(99),ETX1(99),EX1(99),ET1(99),EXX2(99),
     1              ETT2(99),ETX2(99),EXT2(99),EX2(99),ET2(99)
      COMMON /BL31/ DELSQ,EXT1(99)
      COMMON /BL100/ TEEO,$DYNMC
      COMMON /BL101/ DELSD
      COMMON /BL102/ DELOAD
      COMMON /BL103/ MASS(500)
      DIMENSION GEE(4)
C***********************************************************************
      FDIFF(A,B,C)=(-1.5*A+2.*B-.5*C)/DEL
      RS=R(K)
      RR=1./RS
      GA=GAM(K)
      CX=CMXI(K)
      CT=CMT(K)
      DL2=D1*LAM2
      CALL BCB(K,BS,DBS,D,DD)
      CALL PLOAD(K,Z)
      CALL TLOAD(K,Z)
      MAS=MASS(K)
      DO 4 M=1,MNMAX
      I2=K+1+(M-1)*KMAX2
      IK=K+(M-1)*KMAX
      IK1=IK-1
      EN=N(M)
```

```
      ENR=EN*RR
      EMT=MT(M)
      GEE(1) =   (-PX(M)+DT(M)-DL2*GA*OX*EMT)*TDEL*ALCAD
     1   + MAS*(-5.*ZO(1,IZ )+4.*Z2(1,IZ)-Z3(1,IZ))*TDEL/DELSD
      GEE(2) =   (-PT(M)-ENR*TT(M)-DL2*ENR*OT*EMT)*TDEL*ALCAD
     1   + MAS*(-5.*ZO(2,IZ )+4.*Z2(2,IZ)-Z3(2,IZ))*TDEL/DELSD
      GEE(3) =   (-PR(M)-(CX+CT)*TT(M)-DL2*(GA*CMT(M)-(CX*CT-ENR**2)
     1*MT(M)))*TDEL*ALCAD
     1   + MAS*(-5.*ZO(3,IZ )+4.*Z2(3,IZ)-Z3(3,IZ))*TDEL/DELSD
    1 GEE(4) =   MT(M)*TDEL*ALCAD
      IF(ITRMAX.EQ.1) GO TO 5C
      IF(K.GT.1) GO TO 6
      BX2T=BX1(M)
      BT2T=BT1(M)
      BXT2T=BXT1(M)
      BE2T=BE1(M)
      EXX2T=EXX1(M)
      ETT2T=ETT1(M)
      ETX2T=ETX1(M)
      EXT2T=EXT1(M)
      EX2T=EX1(M)
      ET2T=ET1(M)
      DBX= FDIFF(BX2T,BX2(M),BX3(M))
      DBT= FDIFF(BT2T,BT2(M),BT3(M))
      DBXT= FDIFF(BXT2T,BXT2(M),BXT3(M))
      DBE= FDIFF(BE2T,BE2(M),BE3(M))
      DET= FDIFF(ET2T,ET2(M),ET3(M))
      DEX= FDIFF(EX2T,EX2(M),EX3(M))
      DEXX= FDIFF(EXX2T,EXX2(M),EXX3(M))
      DETX= FDIFF(ETX2T,ETX2(M),ETX3(M))
      GO TO 7
    6 IF(K.LT.KMAX) GO TO 8
      BX2T=BX3(M)
      BT2T=BT3(M)
      BXT2T=BXT3(M)
      BE2T=BE3(M)
      EXX2T=EXX3(M)
      ETT2T=ETT3(M)
      ETX2T=ETX3(M)
      EXT2T=EXT3(M)
      EX2T=EX3(M)
      ET2T=ET3(M)
      DBX=-FDIFF(BX2T,BX2(M),BX1(M))
      DBT=-FDIFF(BT2T,BT2(M),BT1(M))
      DBXT=-FDIFF(BXT2T,BXT2(M),BXT1(M))
      DBE=-FDIFF(BE2T,BE2(M),BE1(M))
      DET=-FDIFF(ET2T,ET2(M),ET1(M))
      DEX=-FDIFF(EX2T,EX2(M),EX1(M))
```

```
      DEXX=-FDIFF(EXX2T,EXX2(M),EXX1(M))
      DETX=-FDIFF(ETX2T,ETX2(M),ETX1(M))
      GO TO 7
    8 DBX=TDLI*(BX3(M)-BX1(M))
      DBT=TDLI*(BT3(M)-BT1(M))
      DBE=TDLI*(BE3(M)-BE1(M))
      DBXT=TDLI*(BXT3(M)-BXT1(M))
      DET=TDLI*(ET3(M)-ET1(M))
      DEX=TDLI*(EX3(M)-EX1(M))
      DEXX=TDLI*(EXX3(M)-EXX1(M))
      DETX=TDLI*(ETX3(M)-ETX1(M))
      BX2T=BX2(M)
      BT2T=BT2(M)
      BXT2T=BXT2(M)
      BE2T=BE2(M)
      EXX2T=EXX2(M)
      ETT2T=ETT2(M)
      ETX2T=ETX2(M)
      EXT2T=EXT2(M)
      EX2T=EX2(M)
      ET2T=ET2(M)
    7 GEE(1)=GEE(1)-OSE*(BS*(DBX+DBE+GA*D1*(BX2T-ET2T)+NU*(DBT+DBE)
     1          +ENR*D1*BXT2T)+DBS*(BX2T+BE2T+NU*(BT2T+BE2T))-2.*CX*
     2                 (EXX2T+ETX2T)-ENR*(EX2T+ET2T))*TDEL
      GEE(2)=GEE(2)+OSE*(BS*(ENR*(BT2T+BE2T+NU*(BX2T+BE2T))-D1*
     1      (DBXT+2.*GA*BXT2T))-D1*DBS*BXT2T+2.*CT*(ETT2T+EXT2T)
     2        -(DEX+DET))*TDEL
      GEE(3)=GEE(3)+OSE*(BS*((OX+NU*OT)*(BX2T+BE2T)+(OT+NU*OX)*
     1        (BT2T+BE2T))+2.*(GA*(EXX2T+ETX2T)+DEXX+DETX+ENR*
     2        (EXT2T+ETT2T)))*TDEL
   50 IF(K.GT.1) GO TO 10
      IF(M.GT.1) EL1S(1)=0.0
      DO 20 I=1,4
      GEES(I,M)=GEE(I)
      SUMX=0.
      DO 21 J=1,4
C**********************************************************************
C     FOLLOWING CARD CAUSES A SPECIFIED BOUNDARY CONDITION VALUE TO   *
C     EXISTS ONLY FOR MODE '0'                                         *
C**********************************************************************
      IF (M.NE.1) EL1S(J)=0.
   21 SUMX=SUMX+DL(I,J,M)*EL1S(J)+DG(I,J,M)*GEE(J)+DF(I,J,M)*FFS(J,M)
   20 X(I,IK)=SUMX
      GO TO 4
C    IN FORCE
   10 IF(K.NE.2.OR.(K.EQ.2.AND.IBCINL.GE.0)) GO TO 901
      DO 902 II=1,4
      SUMX=0.
```

```
      DO 903 L=1,4
  903 SUMX=SUMX+CL(II,L,M)*GEE(L)
  902 X(II,IK1)=SUMX
  901 CONTINUE
      DO 11 I=1,4
      SUMX=0.
      DO 12 J=1,4
   12 SUMX=SUMX+DEE(I,J,IK)*GEE(J)-DST(I,J,IK)*X(J,IK1)
   11 X(I,IK)=SUMX
    4 CONTINUE
      RETURN
      END
      SUBROUTINE UPDATE
C*********************************************************************
C        THIS SUBROUTINE UPDATES THE STORAGE LOCATIONS OF THE BETA'S AND
C        ETA'S.  IT IS CALLED IN SUBROUTINE XANDZ AFTER A MERIDIAN      *
C        STATION CHANGE.                                                 *
C*********************************************************************
      COMMON /IBL1/ MNMAX
      COMMON /BL27/ BX3(99),BT3(99),BXT3(99),BE3(99)
      COMMON /BL28/ EXX3(99),ETT3(99),ETX3(99),EXT3(99),EX3(99),ET3(99)
      COMMON /BL29/ BX1(99),BT1(99),BXT1(99),BE1(99),BX2(99),BT2(99),
     1              BXT2(99),BE2(99)
      COMMON /BL30/ EXX1(99),ETT1(99),ETX1(99),EX1(99),ET1(99),EXX2(99),
     1              ETT2(99),ETX2(99),EXT2(99),EX2(99),ET2(99)
      COMMON /BL31/ DELSQ,EXT1(99)
C*********************************************************************
      DO 1 M=1,MNMAX
      BX1(M)=BX2(M)
      BT1(M)=BT2(M)
      BXT1(M)=BXT2(M)
      BE1(M)=BE2(M)
      BX2(M)=BX3(M)
      BT2(M)=BT3(M)
      BXT2(M)=BXT3(M)
      BE2(M)=BE3(M)
      EXX1(M)=EXX2(M)
      ETT1(M)=ETT2(M)
      ETX1(M)=ETX2(M)
      EXT1(M)=EXT2(M)
      EX1(M)=EX2(M)
      ET1(M)=ET2(M)
      EXX2(M)=EXX3(M)
      ETT2(M)=ETT3(M)
      ETX2(M)=ETX3(M)
      EXT2(M)=EXT3(M)
      EX2(M)=EX3(M)
    1 ET2(M)=ET3(M)
```

```
      RETURN
      END
      SUBROUTINE MATINV(A,N,B,M,DETERM,IPIVOT,INDEX,NMAX,ISCALE)
C**********************************************************************
C        THIS SUBROUTINE SOLVES THE MATRIX EQUATION AX=B, WHERE A IS    *
C        A SQUARE COEFFICIENT MATRIX AND B IS A MATRIX OF CONSTANT VEC-*
C        TORS.                                                          *
C        A(INVERSE) IS ALSO OBTAINED AND THE DETERMINANT OF A IS AVAIL-*
C        ABLE.                                                          *
C        THE FOLLOWING MUST BE DIMENSIONED IN THE CALLING PROGRAM:      *
C        IPIVOT(N MAX), INDEX(N MAX,2), A(N MAX,N MAX), B(N MAX,N MAX)  *
C        WHERE:                                                         *
C                 A = NAME OF 2-DIMENSIONAL ARRAY TO BE INVERTED        *
C                 N = ORDER OF A -  1<=N<=NMAX                          *
C                 B = NAME OF 2-DIMENSIONAL ARRAY TO BE MULTIPLIED      *
C                          BY A(INVERSE)                                *
C                 M = NUMBER OF COLUMN VECTORS IN B                     *
C                          NOTE: M = 0 SIGNALS INVERSION ONLY)          *
C                 IPIVOT = TEMPORARY STORAGE BLOCK                      *
C                 INDEX = TEMPORARY STORAGE BLOCK                       *
C                 NMAX = MAXIMUM ORDER OF A (AS DIMENSIONED IN THE      *
C                          CALLING PROGRAM)                             *
C                 DETERM= VALUE OF DETERMINANT AS GIVEN BELOW           *
C                 ISCALE = USED IN FORMULA BELOW                        *
C                                                                       *
C                 DETERMINANT(A) = (10**18)**ISCALE*(DETERM)            *
C                                                                       *
C                 A(INVERSE) IS STORED IN A                             *
C                 A(INVERSE)*B IS STORED IN B                           *
C**********************************************************************
      DIMENSION IPIVOT(N),A(NMAX,N),B(NMAX,M),INDEX(NMAX,2)
      EQUIVALENCE (IROW,JROW), (ICOLUM,JCOLUM), (AMAX, T, SWAP)
C**********************************************************************
C     INITIALIZATION
C
    5 ISCALE=0
    6 R1=10.0**18
    7 R2=1.0/R1
   10 DETERM=1.0
   15 DO 20 J=1,N
   20 IPIVOT(J)=0
   30 DO 550 I=1,N
C
C     SEARCH FOR PIVOT ELEMENT
C
   40 AMAX=0.0
   45 DO 105 J=1,N
   50 IF (IPIVOT(J)-1) 60, 105, 60
```

```
   60 DO 100 K=1,N
   70 IF (IPIVOT(K)-1) 80, 100, 740
   80 IF (ABS(AMAX)-ABS(A(J,K)))85,100,100
   85 IROW=J
   90 ICOLUM=K
   95 AMAX=A(J,K)
  100 CONTINUE
  105 CONTINUE
  110 IPIVOT(ICOLUM)=IPIVOT(ICOLUM)+1
C
C     INTERCHANGE ROWS TO PUT PIVOT ELEMENT ON DIAGONAL
C
  130 IF (IROW-ICOLUM) 140, 260, 140
  140 DETERM=-DETERM
  150 DO 200 L=1,N
  160 SWAP=A(IROW,L)
  170 A(IROW,L)=A(ICOLUM,L)
  200 A(ICOLUM,L)=SWAP
  205 IF(M) 260, 260, 210
  210 DO 250 L=1, M
  220 SWAP=B(IROW,L)
  230 B(IROW,L)=B(ICOLUM,L)
  250 B(ICOLUM,L)=SWAP
  260 INDEX(I,1)=IROW
  270 INDEX(I,2)=ICOLUM
  310 PIVOT=A(ICOLUM,ICOLUM)
C
C     SCALE THE DETERMINANT
C
 1000 PIVOTI=PIVOT
 1005 IF(ABS(DETERM)-R1)1030,1010,1010
 1010 DETERM=DETERM/R1
      ISCALE=ISCALE+1
      IF(ABS(DETERM)-R1)1060,1020,1020
 1020 DETERM=DETERM/R1
      ISCALE=ISCALE+1
      GO TO 1060
 1030 IF(ABS(DETERM)-R2)1040,1040,1060
 1040 DETERM=DETERM*R1
      ISCALE=ISCALE-1
      IF(ABS(DETERM)-R2)1050,1050,1060
 1050 DETERM=DETERM*R1
      ISCALE=ISCALE-1
 1060 IF(ABS(PIVOTI)-R1)1090,1070,1070
 1070 PIVOTI=PIVOTI/R1
      ISCALE=ISCALE+1
      IF(ABS(PIVOTI)-R1)320,1080,1080
 1080 PIVOTI=PIVOTI/R1
```

```
      ISCALE=ISCALE+1
      GO TO 320
 1090 IF(ABS(PIVOTI)-R2)2000,2000,320
 2000 PIVOTI=PIVOTI*R1
      ISCALE=ISCALE-1
      IF(ABS(PIVOTI)-R2)2010,2010,320
 2010 PIVOTI=PIVOTI*R1
      ISCALE=ISCALE-1
  320 DETERM=DETERM*PIVOTI
C
C     DIVIDE PIVOT ROW BY PIVOT ELEMENT
C
  330 A(ICOLUM,ICOLUM)=1.0
  340 DO 350 L=1,N
  350 A(ICOLUM,L)=A(ICOLUM,L)/PIVOT
  355 IF(M) 380, 380, 360
  360 DO 370 L=1,M
  370 B(ICOLUM,L)=B(ICOLUM,L)/PIVOT
C
C     REDUCE NON-PIVOT ROWS
C
  380 DO 550 L1=1,N
  390 IF(L1-ICOLUM) 400, 550, 400
  400 T=A(L1,ICOLUM)
  420 A(L1,ICOLUM)=0.0
  430 DO 450 L=1,N
  450 A(L1,L)=A(L1,L)-A(ICOLUM,L)*T
  455 IF(M) 550, 550, 460
  460 DO 500 L=1,M
  500 B(L1,L)=B(L1,L)-B(ICOLUM,L)*T
  550 CONTINUE
C
C     INTERCHANGE COLUMNS
C
  600 DO 710 I=1,N
  610 L=N+1-I
  620 IF (INDEX(L,1)-INDEX(L,2)) 630, 710, 630
  630 JROW=INDEX(L,1)
  640 JCOLUM=INDEX(L,2)
  650 DO 705 K=1,N
  660 SWAP=A(K,JROW)
  670 A(K,JROW)=A(K,JCOLUM)
  700 A(K,JCOLUM)=SWAP
  705 CONTINUE
  710 CONTINUE
  740 RETURN
      END
      SUBROUTINE INLPOL (Z,PHIXB,PHITB)
```

```
C***********************************************************************
C        THIS SUBROUTINE COMPUTES THE NON-LINEAR TERMS BETA-SUB S,     *
C        -SUB THETA, -SUB S-THETA, ETA-SUB S-S AND -SUB THETA-S AT AN  *
C        INITIAL POLE.                                                  *
C***********************************************************************
      DIMENSION Z(4,1),PHIXB(1),PHITB(1)
      COMMON /IBL1/ MNMAX
      COMMON /IBL3/ MO,M1,M2,M3
      COMMON /IBL12/ KMAX1,KMAX2,NCONV
      COMMON /IBL13/ ITRMAX,LSMAX
     3/IBLJ/ JUMP
      COMMON /BL5/  TT(99),EMT(99),DT(99),DMT(99)
      COMMON /BL6/  SOE,CSE,ALOAD
      COMMON /BL7/  D1,S1
      COMMON /BL11/ OMXI(200),PHEE,TO,T2
      COMMON /BL11A/ PHEN,T2N
      COMMON /BL17/ DEL
      COMMON /BL29/ BX1(99),BT1(99),BXT1(99),BE1(99),BX2(99),BT2(99),
     1              BXT2(99),BE2(99)
      COMMON /BL30/ EXX1(99),ETT1(99),ETX1(99),EX1(99),ET1(99),EXX2(99),
     1              ETT2(99),ETX2(99),EXT2(99),EX2(99),ET2(99)
      COMMON /BL31/ DELSC,EXT1(99)
C***********************************************************************
      DO 1 MN=1,MNMAX
      BX1 (MN)=0.
      BT1 (MN)=0.
      BXT1(MN)=0.
      BE1 (MN)=0.
      EX1 (MN)=0.
      ET1 (MN)=0.
      ETX1(MN)=0.
    1 EXX1(MN)=0.
      IF(M1.EQ.0) RETURN
      I2=2+(M1-1)*KMAX2
      I3=I2+1
      I4=I3+1
      IF(JUMP.EQ.2) GO TO 1000
      PHEE=(1.5*Z(3,I2)-2.*Z(3,I3)+.5*Z(3,I4))/DEL+OMXI(1)*Z(1,I2)
      BET=.5*PHEE**2
      IF(ITRMAX.EQ.1) BET=0.
      T2=0.
      IF(M2.EQ.0) GO TO 2
      CALL BDB(1,B,DB,D,DD)
      I2=2+(M2-1)*KMAX2
      I3=I2+1
      I4=I3+1
      T2=B*D1*((-1.5*Z(1,I2)+2.*Z(1,I3)-.5*Z(1,I4))/DEL+.5*SOE*BET)
      C1=.5*PHEE*T2
```

```
      BX1(M2)=BET
      BT1(M2)=-BET
      BXT1(M2)=-BET
      ETX1(M1)=Q1
      IF(M3.EQ.0) GO TO 2
      EXX1(M3)=Q1
      ETX1(M3)=-Q1
    2 TC=0.
      IF(MO.EQ.0) GO TO 3
      EX1(MO)=BET
      BT1(MO)=BET
      CALL BDB(1,B,DB,D,CC)
      CALL TLOAD(1,Z)
      I2=2+(MO-1)*KMAX2
      I3=I2+1
      I4=I3+1
      TC=B*S1*((-1.5*Z(1,I2)+2.*Z(1,I3)-.5*Z(1,I4))/DEL+CMXI(1)*Z(3,I2)
     1+.5*SOE*BET)-TT(MO)*ALCAD
    3 EXX1(M1)=PHEE*(TO+.5*T2)
      RETURN
 1000 CONTINUE
      PHEE=(1.5*Z(3,I2)-2.*Z(3,I3)+.5*Z(3,I4))/DEL+CMXI(1)*Z(1,I2)
      T2=C.
      IF(M2.EQ.C) GO TO 1002
      CALL BDB(1,B,DB,D,CC)
      I2=2+(M2-1)*KMAX2
      I3=I2+1
      I4=I3+1
      PHX1=PHIXB(KMAX+1)
      PHX2=PHIXB(2*KMAX+1)
      PHEN=(1.5*Z(3,I2-KMAX2)-2.*Z(3,I3-KMAX2)+.5*Z(3,I4-KMAX2))/DEL+
     1CMXI(1)*Z(1,I2-KMAX2)
      BX1(M2)=.5*(PHEE*(PHEE+2.*PHX1)-PHEN*(PHEN+2.*PHX2))
      IF(ITRMAX.EQ.1) BX1(M2)=0.
      BT1(M2)=-BX1(M2)
      BXT1(M2)=-BX1(M2)
      T2=B*D1*((-1.5*Z(1,I2)+2.*Z(1,I3)-.5*Z(1,I4))/DEL+.5*SOE*BX1(M2))
      M2L=M2-1
      BX1(M2L)=PHEE*(PHEN+PHX2)+PHX1*PHEN
      IF(ITRMAX.EQ.1) BX1(M2L)=0.
      BT1(M2L)=-BX1(M2L)
      BXT1(M2L)=BX1(M2L)
      T2N=B*D1*((-1.5*Z(1,I2-KMAX2)+2.*Z(1,I3-KMAX2)-.5*Z(1,I4-KMAX2))
     1  /DEL+.5*SOE*BX1(M2L))
 1002 TC=C.
      IF(MO.EQ.0) GO TO 1003
      BX1(MO)=.5*(PHEE*(PHEE+2.*PHX1)+PHEN*(PHEN+2.*PHX2))
      IF(ITRMAX.EQ.1) BX1(MO)=0.
```

```
      BT1(MO)=BX1(MO)
      CALL BCB(1,B,DB,D,DD)
      CALL TLOAD(1,Z)
      I2=2+(MC-1)*KMAX2
      I3=I2+1
      I4=I3+1
      TC=B*S1*((-1.5*Z(1,I2)+2.*Z(1,I3)-.5*Z(1,I4))/DEL+CMXI(1)*Z(3,I2)
     1+.5*SOE*BX1(MO))-TT(MO)*ALOAD
 1003 IF(ITRMAX.EQ.1) GC TC 1001
      PHSS=PHEE+PHX1
      PHSP=PHEN+PHX2
      M1L=M1-1
      EXX1(M1)=PHSS*T0+.5*(PHSS*T2+PHSP*T2N)
      EXX1(M1L)=PHSP*T0-.5*(PHSP*T2-PHSS*T2N)
      ETX1(M1)=.5*(PHSS*T2+PHSP*T2N)
      ETX1(M1L)=.5*(-PHSP*T2+PHSS*T2N)
      IF(M3.EQ.0) GO TO 1001
      M3L=M3-1
      EXX1(M3)=.5*(PHSS*T2-PHSP*T2N)
      EXX1(M3L)=.5*(PHSS*T2N+PHSP*T2)
      ETX1(M3)=.5*(-PHSS*T2+PHSP*T2)
      ETX1(M3L)=.5*(-PHSP*T2-PHSS*T2N)
 1001 CONTINUE
      RETURN
      END
      SUBROUTINE ABC
C***********************************************************************
C        THIS SUBROUTINE COMPUTES THE ELEMENTS OF THE A, BEE, AND C    *
C        MATRICES.                                                     *
C***********************************************************************
      COMMON /BL1/  A(4,4),BEE(4,4),C(4,4)
      COMMON /BL12/ TDLI,TDEL
      COMMON /BL17/ DEL
      COMMON /BL25/ E(4,4),F(4,4),G(4,4)
C***********************************************************************
      D2=2./DEL
      DO 1 I=1,4
      DO 1 J=1,4
      DEIJ=D2*E(I,J)
      FIJ=F(I,J)
      BEE(I,J)=-2.*DEIJ+TDEL*G(I,J)
      C(I,J)=DEIJ-FIJ
    1 A(I,J)=DEIJ+FIJ
      RETURN
      END
      SUBROUTINE PANDD(K,MN,P,DEE,DST,X)
C***********************************************************************
C        THIS SUBROUTINE COMPUTES THE ELEMENTS OF THE P, P-BAR, AND    *
```

```
C         P-HAT MATRICES FOR EACH MERIDIAN STATION K AND FOURIER MODE MN*00008400
C         THESE MATRICES ARE COMPUTED AND SAVED BECAUSE THEY DO NOT     *00008410
C         CHANGE DURING EITHER THE ITERATION PROCEDURE OR THE LOAD INCRE-00008420
C         MENT PROCEDURE - AS THEY ARE A FUNCTION OF THE SHELL'S INITIAL*00008430
C         GEOMETRY AND STIFFNESS.                                       *00008440
C***********************************************************************00008450
      DIMENSION P(4,4,1),CEE(4,4,1),DST(4,4,1),X(4,1)                   00008460
      COMMON /IBL4/ KMAX,KL                                             00008470
      COMMON /BL1/  A(4,4),BEE(4,4),C(4,4)                              00008480
      COMMON /BL4/  ZF1M(4,4,99),ZF2M(4,4,99),
     1              ZF3M(4,4,99),ZF4M(4,4,99)
      DIMENSION TM(4,4),IPIVOT(4),INDEX(4,2),X2(4)                      00008510
C***********************************************************************00008520
      IKL=K+KMAX*(MN-1)                                                 00008530
      KLI=IKL-1                                                         00008540
      DO 1 I=1,4                                                        00008550
      DO 1 J=1,4                                                        00008560
      SUM=0.                                                            00008570
      DO 2 L=1,4                                                        00008580
    2 SUM=SUM+C(I,L)*P(L,J,KLI)                                         00008590
    1 TM(I,J)=BEE(I,J)-SUM                                              00008600
      CALL MATINV(TM,4,X2,0,DETERM,IPIVOT,INDEX,4,ISCALE)               00008610
      DO 5 I=1,4                                                        00008620
      DO 5 J=1,4                                                        00008630
      SUMA=0.                                                           00008640
      SUMC=0.                                                           00008650
      DO 6 L=1,4                                                        00008660
      SUMA=SUMA+TM(I,L)*A(L,J)                                          00008670
    6 SUMC=SUMC+TM(I,L)*C(L,J)                                          00008680
      P(I,J,IKL)=SUMA                                                   00008690
      CEE(I,J,IKL)=TM(I,J)                                              00008700
    5 DST(I,J,IKL)=SUMC                                                 00008710
      RETURN                                                            00008720
      END                                                               00008730
      SUBROUTINE FNLPOL (Z,PHIXB,PHITB)                                 00008740
C***********************************************************************00008750
C         THIS SUBROUTINE COMPUTES THE NON-LINEAR TERMS BETA-SUB S,     *00008760
C         -SUB THETA, -SUB S-THETA, ETA-SUB S-S, AND -SUB THETA-S AT A  *00008770
C         FINAL POLE.                                                   *00008780
C***********************************************************************00008790
      DIMENSION Z(4,1),PHIXB(1),PHITB(1)                                00008800
      COMMON /IBL1/ MNMAX                                               00008810
      COMMON /IBL3/ MO,M1,M2,M3                                         00008820
      COMMON /IBL4/ KMAX,KL                                             00008830
      COMMON /IBL12/ KMAX1,KMAX2,NCONV                                  00008840
      COMMON /IBL13/ ITRMAX,LSMAX                                       00008850
      COMMON /IBLJ/ JUMP                                                00008860
      COMMON /BL5/  TT(99),EMT(99),DT(99),DMT(99)                       00008870
```

```
      COMMON /BL6/  SOE,CSE,ALOAD
      COMMON /BL7/  D1,S1
      COMMON /BL11/ OMXI(500),PHEE,TO,T2
      COMMON /BL11A/ PHEN,T2N
      COMMON /BL17/ DEL
      COMMON /BL27/ BX3(99),BT3(99),BXT3(99),BE3(99)
      COMMON /BL28/ EXX3(99),ETT3(99),ETX3(99),EXT3(99),EX3(99),ET3(99)
C***********************************************************************
      DO 1 MN=1,MNMAX
      BX3 (MN)=0.
      BT3 (MN)=0.
      BXT3(MN)=0.
      BE3 (MN)=0.
      EX3 (MN)=0.
      ET3 (MN)=0.
      ETX3(MN)=0.
    1 EXX3(MN)=0.
      CALL BCB(KMAX,B,DB,D,DB)
      IF(M1.EQ.0) RETURN
      KM=KMAX1+(M1-1)*KMAX2
      KM1=KM-1
      KM2=KM-2
      PHEE=-(1.5*Z(3,KM)-2.*Z(3,KM1)+.5*Z(3,KM2))/DEL+OMXI(KMAX)*Z(1,KM)
      IF(JUMP.EQ.2) GO TO 1000
      BET=.5*PHEE**2
      IF(ITRMAX.EQ.1) BET=0.
      T2=0.
      IF(M2.EQ.0) GO TO 2
      KM=KMAX1+(M2-1)*KMAX2
      KM1=KM-1
      KM2=KM-2
      T2=B*D1*((1.5*Z(1,KM)-2.*Z(1,KM1)+.5*Z(1,KM2))/DEL+.5*SCE*BET)
      Q1=.5*PHEE*T2
      BX3(M2)=BET
      ET3(M2)=-BET
      EXT3(M2)= BET
      ETX3(M1)=Q1
      IF(M3.EQ.0) GO TO 2
      EXX3(M3)=Q1
      ETX3(M3)=-Q1
    2 TC=0.
      IF(MO.EQ.0) GO TO 3
      CALL TLOAD(KMAX,Z)
      KM=KMAX1+(MO-1)*KMAX2
      KM1=KM-1
      KM2=KM-2
      BX3(MO)=BET
      BT3(MO)=BET
```

```
      TC=B*S1*((1.5*Z(1,KM)-2.*Z(1,KM1)+.5*Z(1,KM2))/DEL+GMXI(KMAX)*
     1Z(3,KM)+.5*SOE*BE1)-TT(MO)*ALOAD
    3 EXX3(M1)=PHEE*(TO+.5*T2)
      RETURN
C*********************************************************************
 1000 CONTINUE
      T2=C.
      IF(M2.EQ.0) GO TO 1002
      KM=KMAX1+(M2-1)*KMAX2
      KM1=KM-1
      KM2=KM-2
      J=KMAX*2
      I=J+KMAX
      PHX1=PHIXB(J)
      PHX2=PHIXB(I)
      PHEN=-(1.5*Z(3,KM-KMAX2)-2.*Z(3,KM1-KMAX2)+.5*Z(3,KM2-KMAX2))/DEL
     1+CMXI(KMAX)*Z(1,KM-KMAX2)
      BX3(M2)=.5*(PHEE*(PHEE+2.*PHX1)-PHEN*(PHEN+2.*PHX2))
      IF(ITRMAX.EQ.1) BX3(M2)=0.
      BT3(M2)=-BX3(M2)
      BXT3(M2)=BX3(M2)
      M2L=M2-1
      BX3(M2L)=PHEE*(PHEN+PHX2)+PHX1*PHEN
      IF(ITRMAX.EQ.1) BX3(M2L)=0.
      BT3(M2L)=-BX3(M2L)
      BXT3(M2L)=-BX3(M2L)
      T2=B*D1*((1.5*Z(1,KM)-2.*Z(1,KM1)+.5*Z(1,KM2))/DEL+.5*SCE*BX3(M2))
      T2N=B*D1*((1.5*Z(1,KM-KMAX2)-2.*Z(1,KM1-KMAX2)+.5*Z(1,KM2-KMAX2))
     1  /DEL+.5*SOE*BX3(M2L))
 1002 TC=C.
      IF(MO.EQ.0) GO TO 1003
      CALL TLOAD(KMAX,Z)
      KM=KMAX1+(MO-1)*KMAX2
      KM1=KM-1
      KM2=KM-2
      BX3(MO)=.5*(PHEE*(PHEE+2.*PHX1)+PHEN*(PHEN+2.*PHX2))
      IF(ITRMAX.EQ.1) BX3(MO)=0.
      BT3(MO)=BX3(MO)
      TC=B*S1*((1.5*Z(1,KM)-2.*Z(1,KM1)+.5*Z(1,KM2))/DEL+CMXI(KMAX)*
     1  Z(3,KM)+.5*SOE*BX3(MO))-TT(MO)*ALOAD
 1003 IF(ITRMAX.EQ.1) GO TO 1001
      PHSS=PHEE+PHX1
      PHSP=PHEN+PHX2
      M1L=M1-1
      EXX3(M1)=PHSS*TO+.5*(PHSS*T2+PHSP*T2N)
      EXX3(M1L)=PHSP*TO-.5*(PHSP*T2-PHSS*T2N)
      ETX3(M1)=.5*(PHSS*T2+PHSP*T2N)
      ETX3(M1L)=.5*(-PHSP*T2+PHSS*T2N)
```

```
      IF(M3.EQ.0) GO TO 1001
      M3L=M3-1
      EXX3(M3)=.5*(PHSS*T2-PHSP*T2N)
      EXX3(M3L)=.5*(PHSS*T2N+PHSP*T2)
      ETX3(M3)=.5*(-PHSS*T2+PHSP*T2N)
      ETX3(M3L)=.5*(-PHSP*T2-PHSS*T2N)
 1001 CONTINUE
      RETURN
      END
      SUBROUTINE PHIBET(K,Z,IS,JS,ID,JD,PHIXB,PHITB)
C***********************************************************************
C        THIS SUBROUTINE CALCULATES THE PHI'S AND CARRIES OUT THE      *
C        MULTIPLYING AND SUMMATION PROCEDURE FOR COMPUTING THE BETA    *
C        NON-LINEAR TERMS FOR A GIVEN MERIDIONAL STATION K.  THE ARRAYS*
C        IS, JS, ID, JS,IJS, MAXS, MAXC, MAXSY ARE PREPARED IN SUB-    *
C        ROUTINE MODES AND USED HERE.                                  *
C***********************************************************************
      DIMENSION Z(4,1),IS(99,1),JS(99,1),ID(99,1),JD(99,1),PHIXB(1),
     1PHITB(1)
      COMMON /IBL1/ MNMAX
      COMMON /IBL2/ N(99),MNINIT
      COMMON /IBL4/ KMAX,KL
      COMMON /IBL7/ MNMAXO,MAXD(99),MAXS(99),MAXSY(99),IJS(99)
      COMMON /IBL12/ KMAX1,KMAX2,NCONV
      COMMON /IBL13/ ITRMAX,LSMAX
      COMMON /IBLJ/ JUMP
      COMMON /BL6/  SOE,CSE,ALOAD
      COMMON /BL8/  R(500),GAM(500),OMT(500)
      COMMON /BL10/ PHIX(99),PHIT(99),PHI(99)
      COMMON /BL11/ OMXI(500),PHEE,TO,T2
      COMMON /BL12/ TDLI,TDEL
      COMMON /BL15/ NU,U1(99),V1(99),W1(99),V2(99),U2(99),W2(99),U3(99),
     1              V3(99),W3(99)
      COMMON /BL27/ BX3(99),BT3(99),BXT3(99),BE3(99)
      COMMON /BLPHS/ PHX(99),PHT(99)
C***********************************************************************
      CX=OMXI(K)
      CT=OMT(K)
      RRA=1./R(K)
      GA=GAM(K)
      KP2=K+2
      DO 1 M=1,MNMAXO
      EN=N(M)
      IK=KP2+(M-1)*KMAX2
      U3(M)=Z(1,IK)
      V3(M)=Z(2,IK)
      W3(M)=Z(3,IK)
      PHIX(M)=-TDLI*(W3(M)-W1(M))+CX*U2(M)
```

```
      PHIT(M)=EN*W2(M)*RRA+V2(M)*CT
    1 PHI(M)=(TDLI*(V3(M)-V1(M))+GA*V2(M)+EN*U2(M)*RRA)*.5
      IF(ITRMAX.EQ.1) RETURN
      IF(JUMP.EQ.2) GO TO 1111
      DO 9 M=1,MNMAX
      SMC=0.
      SMT=0.
      SMR=0.
      SMF=0.
      IF(N(M).EQ.0) GO TO 20
      MAXL=MAXS(M)
      IF(MAXL.EQ.0) GO TO 2
      DO 3 L=1,MAXL
      I=IS(L,M)
      J=JS(L,M)
      SMC=SMO+PHIX(I)*PHIX(J)
      SMT=SMT-PHIT(I)*PHIT(J)
      SMR=SMR+PHIX(I)*PHIT(J)+PHIX(J)*PHIT(I)
    3 SMF=SMF-PHI(I)*PHI(J)
    2 MAXL=MAXD(M)
      IF(MAXL.EQ.0) GO TO 4
      DO 5 L=1,MAXL
      I=ID(L,M)
      J=JD(L,M)
      SMC=SMO+PHIX(I)*PHIX(J)
      SMT=SMT+PHIT(I)*PHIT(J)
      SMR=SMR-PHIX(I)*PHIT(J)+PHIX(J)*PHIT(I)
    5 SMF=SMF+PHI(I)*PHI(J)
    4 IF(MAXSY(M).EQ.0) GC TO 10
      I=IJS(M)
      SMC=SMO+PHIX(I)**2/2.
      SMT=SMT-PHIT(I)**2/2.
      SMR=(SMR+FHIX(I)*PHIT(I))
      SMF=SMF-PHI(I)**2/2.
      GC TO 10
   20 DO 21 L=1,MNMAXO
      SMC=SMO+PHIX(L)**2
      SMT=SMT+PHIT(L)**2
   21 SMF=SMF+PHI(L)**2
      IF(M.GT.MNMAXO) GC TO 11
      SMC=SMO+PHIX(M)**2
   11 BX3(M)=SMO*.5
      BT3(M)=SMT*.5
      BE3(M)=SMF*.5
      BXT3(M)=0.
      GC TO 9
   10 EX3(M)=SMO
      BT3(M)=SMT
```

```
      EXT3(M)=SMR*.5
      BE3(M)=SMF
    9 CONTINUE
      RETURN
C************************************************************************
C                                                                       *
C                                                                       *
C     THIS SECTION HANDLES GENERAL LOADING AND IMPERFECTIONS            *
C                                                                       *
C                                                                       *
C************************************************************************
 1111 CONTINUE
      DO 60 M=1,MNMAXO
      KP=K+(M-1)*KMAX
      PHX(M)=PHIXB(KP)
   60 PHT(M)=PHITB(KP)
      DO 49 M=1,MNMAX
      SMC=0.
      SMT=0.
      SMR=0.
      SMF=0.
C************************************************************************
C     TEST FOR ASYMMETRIC MODE                                          *
C************************************************************************
      IF (N(M).LT.0) GO TO 101
C************************************************************************
C                                                                       *
C     THIS SECTION HANDLES SYMMETRIC MODE COMBINATIONS ONLY             *
C                                                                       *
C************************************************************************
C     TEST FOR ZERO- TH MODE                                            *
C************************************************************************
      IF(N(M).EQ.0) GO TO  420
      MAXL=MAXS(M)
C************************************************************************
C     TEST FOR PRESENCE OF SYMMETRIC SUMMATION COMBINATIONS             *
C************************************************************************
      IF(MAXL.EQ.0) GO TO 42
C************************************************************************
C     SET UP -COUPLING- MODES- INDICES AND TEST FOR MODE 1
C************************************************************************
      DO 43 L=1,MAXL
      I=IS(L,M)
      J=JS(L,M)
      II=I-1
      JJ=J-1
      IF (I.EQ.1) GO TO 43
C************************************************************************
```

```
C      COMPILE SUMS FOR SYMMETRIC SUMMATION COMBINATION MODES           *
C***********************************************************************
      SMC=SMC+PHIX(I)*PHIX(J)
     1    +PHX(I)*PHIX(J)+PHX(J)*PHIX(I)
     2         -PHIX(II)*(PHIX(JJ)+PHX(JJ))-PHIX(JJ)*PHX(II)
      SMT=SMT-PHIT(I)*PHIT(J)
     1    -PHT(I)*PHIT(J)-PHT(J)*PHIT(I)
     2         +PHIT(II)*(PHIT(JJ)+PHT(JJ))+PHIT(JJ)*PHT(II)
      SMR=SMR+PHIX(I)*PHIT(J)+PHIX(J)*PHIT(I)
     1    +PHX(I)*PHIT(J)+PHX(J)*PHIT(I)+PHIX(I)*PHT(J)
     2    +PHIX(J)*PHT(I)
     3         +PHIX(II)*(PHIT(JJ)+PHT(JJ))+PHIX(JJ)*(PHIT(II)
     4         +PHT(II))+PHX(II)*PHIT(JJ)+PHX(JJ)*PHIT(II)
      SMF=SMF-PHI(I)*PHI(J)
     1         +PHI(II)*PHI(JJ)
   43 CONTINUE
   42 MAXL=MAXD(M)
C***********************************************************************
C      TEST FOR PRESENCE OF SYMMETRIC DIFFERENCE COMBINATIONS           *
C***********************************************************************
      IF(MAXL.EQ.0) GO TO 44
C***********************************************************************
C      SET UP -COUPLING- MODES- INDICES AND TEST FOR MODE 1
C***********************************************************************
      DO 45 L=1,MAXL
      I=ID(L,M)
      J=JD(L,M)
      II=I-1
      JJ=J-1
      IF (J.EQ.1) GO TO 442
C***********************************************************************
C      COMPILE SUMS FOR SYMMETRIC DIFFERENCE COMBINATION MODES          *
C***********************************************************************
      SMC=SMO+PHIX(I)*PHIX(J)
     1    +PHX(I)*PHIX(J)+PHX(J)*PHIX(I)
     2         +PHIX(II)*(PHIX(JJ)+PHX(JJ))+PHIX(JJ)*PHX(II)
      SMT=SMT+PHIT(I)*PHIT(J)
     1    +PHT(I)*PHIT(J)+PHT(J)*PHIT(I)
     2         +PHIT(II)*(PHIT(JJ)+PHT(JJ))+PHIT(JJ)*PHT(II)
      SMR=SMR-PHIX(I)*PHIT(J)+PHIX(J)*PHIT(I)
     1    -PHX(I)*PHIT(J)+PHX(J)*PHIT(I)-PHIX(I)*PHT(J)
     2    +PHIX(J)*PHT(I)
     3         +PHIX(II)*(PHIT(JJ)+PHT(JJ))-PHIX(JJ)*(PHIT(II)
     4         +PHT(II))+PHX(II)*PHIT(JJ)-PHX(JJ)*PHIT(II)
      SMF=SMF+PHI(I)*PHI(J)
     1         +PHI(II)*PHI(JJ)
      GO TO 45
C***********************************************************************
```

```
C      EXECUTE BELOW IF J=1 IN DIFFERENCE COMBINATIONS                    *
C***********************************************************************
  442 SMC=SMC+(PHIX(1)*PHIX(I)+PHX(1)*PHIX(I)+PHX(I)*PHIX(1))*2.0
      SMT=SMT+(PHIT(1)*(PHIT(II)+PHT(II))+PHIT(II)*PHT(1))*2.0
      SMR=SMR+(PHIX(1)*(PHIT(I)+PHT(I))+PHX(1)*PHIT(I)
     1       +PHIT(1)*(PHIX(II)+PHX(II))+PHT(1)*PHIX(II))*2.0
      SMF=SMF+(PHI(1)*PHI(II))*2.0
   45 CONTINUE
C***********************************************************************
C      TEST FOR PRESENCE OF SAME-INDEX COMBINATION                        *
C***********************************************************************
   44 IF(MAXSY(M).EQ.0) GO TO 410
C***********************************************************************
C      SET UP INDICES AND COMPILE SUMS                                    *
C***********************************************************************
      I=IJS(M)
      II=I-1
      SMC=SMO+PHIX(I)**2/2.
     1    +PHX(I)*PHIX(I)
     2       -PHIX(II)**2/2.0-PHIX(II)*PHX(II)
      SMT=SMT-PHIT(I)**2/2.
     1    -PHT(I)*PHIT(I)
     2       +PHIT(II)**2/2.0+PHIT(II)*PHT(II)
      SMR= SMR+PHIX(I)*PHIT(I)
     1    +PHX(I)*PHIT(I)+PHIX(I)*PHT(I)
     2       +PHIX(II)*(PHIT(II)+PHT(II))+PHX(II)*PHIT(II)
      SMF=SMF-PHI(I)**2/2.
     2       +PHI(II)**2/2.0
      GO TO 410
C***********************************************************************
C      THIS SECTION HANDLES CASES WHERE MODE 0 IS INCLUDED                *
C***********************************************************************
  420 DO 421 L=1,MNMAXO
      SMC=SMO+PHIX(L)**2
     1    +2.0*PHX(L)*PHIX(L)
      SMT=SMT+PHIT(L)**2
     1    +2.0*PHT(L)*PHIT(L)
  421 SMF=SMF+PHI(L)**2
      SMC=SMC+PHIX(M)**2
     1    +2.0*PHX(M)*PHIX(M)
      SMF=SMF+PHI(1)**2
      SMT=SMT+PHIT(1)*(PHIT(1)+2.0*PHT(1))
      SMR=SMR+(PHIX(1)*(PHIT(1)+PHT(1))+PHX(1)*PHIT(1))*2.0
      DO 122 L=2,MNMAXO,2
  122 SMR=SMR+PHIX(L)*(PHIT(L+1)+PHT(L+1))+PHX(L)*PHIT(L+1)
      DO 121 L=3,MNMAXO,2
  121 SMR=SMR+PHIX(L)*(PHIT(L-1)+PHT(L-1))+PHX(L)*PHIT(L-1)
C***********************************************************************
```

```
C     COMPILE BETA TERMS FOR THIS SPECIAL CASE                         *
C***********************************************************************
      BX3(M)=SMO*.5
      BT3(M)=SMT*.5
      BE3(M)=SMF*.5
      BXT3(1)=SMR*0.5
      GO TO 49
C***********************************************************************
C                                                                      *
C     THIS SECTION HANDLES ASYMMETRIC MODES                            *
C                                                                      *
C***********************************************************************
  101 MP=M+1
      MAXL=MAXS(MP)
C***********************************************************************
C     TEST FOR PRESENCE OF SUMMATION COMBINATIONS                      *
C***********************************************************************
      IF (MAXL.EQ.0) GO TO 102
C***********************************************************************
C     SET UP INDICES FOR SUMMATION COMBINATIONS                        *
C***********************************************************************
      DO 103 L=1,MAXL
      I=IS(L,MP)
      J=JS(L,MP)
      II=I-1
      JJ=J-1
C***********************************************************************
C     TEST FOR MODE 1 AND COMPILE SUMS                                 *
C***********************************************************************
      IF (I.EQ.1) GO TO 103
      SMO=SMO+PHIX(I)*(PHIX(JJ)+PHX(JJ))+PHIX(J)*(PHIX(II)+
     1    PHX(II))+PHIX(II)*PHX(J)+PHIX(JJ)*PHX(I)
      SMT=SMT+PHIT(I)*(PHIT(JJ)+PHT(JJ))+PHIT(J)*(PHIT(II)+
     1    PHT(II))+PHIT(II)*PHT(J)+PHIT(JJ)*PHT(I)
      SMR=SMR+PHIX(I)*(PHIT(JJ)+PHT(JJ))+PHIX(J)*(PHIT(II)+
     1    PHT(II))-PHIX(II)*(PHIT(J)+PHT(J))-PHIX(JJ)*
     2    (PHIT(I)+PHT(I))+PHX(I)*PHIT(JJ)+PHX(J)*PHIT(II)-
     3    PHX(II)*PHIT(J)-PHX(JJ)*PHIT(I)
      SMF=SMF+PHI(I)*PHI(JJ)+PHI(J)*PHI(II)
  103 CONTINUE
C***********************************************************************
C     TEST FOR PRESENCE OF DIFFERENCE COMBINATIONS                     *
C***********************************************************************
  102 MAXL=MAXD(MP)
      IF (MAXL.EQ.0) GO TO 104
C***********************************************************************
C     SET UP INDICES FOR DIFFERENCE COMBINATIONS                       *
C***********************************************************************
```

```
      DC 105 L=1,MAXL
      I=IC(L,MP)
      J=JC(L,MP)
      I  I-1
          -1
C**        ****************************************************************
C          FOR MODE 1 ANC CCMPILE SUMS                                    *
C*****  ***.************************************************************
      IF (J.EC.1) GO TO 123
      SMC=SMO-PHIX(I)*(PHIX(JJ)+PHX(JJ))+PHIX(J)*(PHIX(II)+
     1    PHX(II))+PHIX(II)*PHX(J)-PHIX(JJ)*PHX(I)
      SMT=SMT+PHIT(I)*(PHIT(JJ)+PHT(JJ))-PHIT(J)*(PHIT(II)+
     1    PHT(II))-PHIT(II)*PHT(J)+PHIT(JJ)*PHT(I)
      SMR=SMR+PHIX(I)*(PHIT(JJ)+PHT(JJ))+PHIX(J)*(PHIT(II)+
     1    PHT(II))+PHIX(II)*(PHIT(J)+PHT(J))+PHIX(JJ)*
     2    (PHIT(I)+PHT(I))+PHX(I)*PHIT(JJ)+PHX(J)*PHIT(II)+
     3    PHX(II)*PHIT(J)+PHX(JJ)*PHIT(I)
      SMF=SMF+PHI(I)*PHI(JJ)-PHI(J)*PHI(II)
      GC TO 1C5
C***********************************************************************
C     EXECUTE BELOW IF J=I IN DIFFERENCE COMBINATIONS                   *
C***********************************************************************
  123 SMC=SMO+(PHIX(1)*(PHIX(II)+PHX(II))+PHIX(II)*PHX(1))*2.0
      SMT=SMT+(PHIT(I)*(PHIT(1)+PHT(1))+PHIT(1)*PHT(II))*2.C
      SMR=SMR+(PHIX(1)*(PHIT(II)+PHT(II))+PHIX(II)*(PHIT(1)+
     1    PHT(1))+PHX(1)*PHIT(II)+PHX(II)*PHIT(1))*2.0
      SMF=SMF+(PHI(I)*PHI(1))*2.0
  1C5 CCNTINUE
C***********************************************************************
C     TEST FOR PRESENCE OF SAME-INDEX CCMBINATICN                       *
C***********************************************************************
  1C4 IF (MAXSY(MP).EQ.0) GC TO 410
C***********************************************************************
C     SET UP INDICES AND CCMPILE SUMS                                   *
C***********************************************************************
      I=IJS(MP)
      II=I-1
      SMC=SMO+PHIX(I)*(PHIX(II)+PHX(II))+PHIX(II)*PHX(I)
      SMT=SMT+PHIT(I)*(PHIT(II)+PHT(II))+PHIT(II)*PHT(I)
      SMR=SMR+PHIX(I)*(PHIT(II)+PHT(II))-PHIX(II)*(PHIT(I)+
     1    PHT(I))+PHX(I)*PHIT(II)-PHX(II)*PHIT(I)
      SMF=SMF+PHI(I)*PHI(II)
C***********************************************************************
C     CCMPILE BETA TERMS                                                *
C***********************************************************************
  410 BX3(M)=SMO
      BT3(M)=SMT
      BXT3(M)=SMR*.5
```

```
      BE3(M)=SMF
   49 CONTINUE
      RETURN
      END
      SUBROUTINE HJ(K,MN)
C***********************************************************************
C         THIS SUBROUTINE COMPUTES THE ELEMENTS OF THE H AND JAY       *
C         MATRICES FOR BOTH BOUNDARIES OF THE SHELL                    *
C***********************************************************************
      REAL L2,LAM2,LSD1N,        LSD18,JAY,NU
      COMMON /IBL2/ N(99),MNINIT
      COMMON /IBL4/ KMAX,KL
      COMMON /BL8/  R(500),GAM(500),OMT(500)
      COMMON /BL11/ OMXI(500),PHEE,T0,T2
      COMMON /BL14/ LAM2,LSD18,LSD1N
      COMMON /BL15/ NU,U1(99),V1(99),W1(99),V2(99),L2(99),W2(99),U3(99),
     1              V3(99),W3(99)
      COMMON /BL17/ DEL
      COMMON /BL20/ DEOMX(500)
      COMMON /BL23/ JAY(4,4),H(4,4)
      EQUIVALENCE(L2,LAM2)
C***********************************************************************
      CALL BCB(K,B,DB,D,DD)
      YAH=1.
      IF(K.EQ.1.OR.K.EQ.KMAX)YAH=2.
      D1=(1.-NU)
      GA=GAM(K)
      OX=OMXI(K)
      RA=R(K)
      EN=N(MN)
      ENR=EN/RA
      REG=0.
      IF(YAH.EQ.2.) REG=1.
      OT=OMT(K)
      OXT=3.*OMXI(K)-OMT(K)
      OTX=3.*OMT(K)-OMXI(K)
      DL=D*L2*D1*ENR
      H(1,1)=B
      H(1,2)=0.
      H(1,3)=0.
      H(1,4)=0.
      H(2,1)=0.
      H(2,2)=B*D1/2.+L2*D*D1/8.*OTX**2*REG
      H(2,3)=DL/2.*OTX*REG
      H(2,4)=0.
      H(3,1)=0.
      H(3,2)=DL*OTX*YAH/4.
      ENR2=ENR**2
```

```
      H(3,3)=L2*D*D1*(YAH*ENR2+(1.+NU)*GA**2)
      GA2=GA**2
      H(3,4)=L2
      H(4,1)=0.
      H(4,2)=0.
      H(4,3)=-1.
      H(4,4)=0.
      JAY(1,1)=NU*GA*B
      JAY(1,2)=NU*B*ENR
      JAY(1,3)=B*(OX+NU*CT)
      JAY(1,4)=0.
      JAY(2,1)=-B*D1*ENR/2.-DL/8.*OXT*CTX*REG
      JAY(2,2)=-GA*H(2,2)
      JAY(2,3)=-GA*H(2,3)
      JAY(2,4)=0.
      JAY(3,1)=-L2*D*D1*((1.+NU)*GA2*OX+ENR2/4.*CXT*YAH)
      JAY(3,2)=-GA*DL/2.*(     2.*CT*(1.+NU)+OTX/2.*YAH)
      JAY(3,3)=-L2*D*D1*(1.+NU+YAH)*GA*ENR2
      JAY(3,4)=L2*D1*GA
      JAY(4,1)=OX
      JAY(4,2)=0.
      JAY(4,3)=0.
      JAY(4,4)=0.
      DO 1 I=1,4
      DO 1 J=1,4
    1 H(I,J)=H(I,J)/2./DEL
      RETURN
      END
      SUBROUTINE EFG(K,MN,ZO,Z2,Z3)
C*************************************************************************
C         THIS SUBROUTINE PREPARES THE ELEMENTS OF THE E, F, AND G       *
C         FOR EACH MERIDIAN STATION K AND FOR EACH FOURIER MODE, MN.     *
C*************************************************************************
      IMPLICIT LOGICAL*1 ($)
      REAL NU,N,     LAM2,LSD18,LSD1N,MASS,MAS
      DIMENSION ZO(4,1),Z2(4,1),Z3(4,1)
      COMMON /IBL2/NN(99),MNINIT
      COMMON /BL8/  R(500),GAM(500),OMT(500)
      COMMON /BL11/ OMXI(500),PHEE,TO,T2
      COMMON /BL14/ LAM2,LSD18,LSD1N
      COMMON /BL15/ NU,U1(99),V1(99),W1(99),V2(99),U2(99),W2(99),U3(99),
     1              V3(99),W3(99)
      COMMON /BL20/ DEOMX(500)
      COMMON /BL25/ E(4,4),F(4,4),G(4,4)
      COMMON /BL100/ TEEC,$DYNMC
      COMMON /BL101/ DELSD
      COMMON /BL102/ DELOAD
      COMMON /BL103/ MASS(500)
```

```
C*****************************************************************************
      N=NN(MN)
      MAS=MASS(K)
      CALL BCB(K,B,DB,D,CC)
      E(1,1)=B
      E(1,2)=0.
      E(1,3)=0.
      E(1,4)=0.
      E(2,1)=0.
      C1=(1.-NU)
      RA=R(K)
      GA=GAM(K)
      CX=CMXI(K)
      CT=CMT(K)
      DEX=DEDMX(K)
      REX=(3.*OT-OX)
      GA2=GA**2
      RXE=(3.*OX-OT)
      CTX=OT*CX
      CNLR=LAM2*D*N*D1/(2.*RA)
      CCNLR=CNLR*DD/D
      E(2,2)=B*C1/2.+LAM2*D*D1*REX**2/8.
      E(2,3)=CNLR*REX
      E(2,4)=0.
      E(3,1)=0.
      E(3,2)=E(2,3)
      RAN=(N/RA)**2
      E(3,3)=LAM2*D*D1*(2.*RAN+(1.+NU)*GA2)
      E(3,4)=LAM2
      E(4,1)=0.
      E(4,2)=0.
      E(4,3)=-D
      E(4,4)=0.
      F(1,1)=GA*B+DB
      F(1,2)=(1.+NU)*B*N/(2.*RA)+CNLR*REX*RXE/4.
      F(1,3)=B*(OX+NU*OT)+LAM2*D*D1*((1.+NU)*GA2*CX+RAN*RXE/2.)
      F(1,4)=LAM2*OX
      F(2,1)=-F(1,2)
      F(2,2)=(D1/2.)*(GA*B+DB)-(LAM2*C*C1*REX/8.)*(2.*DEX-GA*(5.*OX
     1-3.*OT))+LAM2*DD*C1*REX**2/8.
      F(2,3)=CNLR*(2.*(1.+NU)*GA*OT-DEX+3.*GA*(CX-CT))+CCNLR*REX
      F(2,4)=0.
      F(3,1)=-F(1,3)
      F(3,2)=CNLR*(3.*GA*OX-GA*OT*(5.+2.*NU)-DEX)+CCNLR*REX
      F(3,3)=-LAM2*D*D1*((1.+NU)*(2.*GA*OX*OT+GA**3)+2.*GA*RAN)
     1+LAM2*CC*D1*((1.+NU)*GA2+2.*RAN)
      F(3,4)=LAM2*GA*(2.-NU)
      F(4,1)=C*CX
```

```
      F(4,2)=0.
      F(4,3)=-D*NU*GA
      F(4,4)=0.
      G(1,1)=NU*DB*GA-NU*B*OTX-B*GA2-D1*B*RAN/2.-LAM2*D*D1*((1.+NU)*GA2*
     1CX**2+RXE**2*RAN/8.)
     2   - 2.*MAS/DELSD
      G(1,2)=NU*N*DB/RA-(3.-NU)/(2.*RA)*GA*B*N-DNLR*2.*GA*(REX*RXE/8.
     1+(1.+NU)*OTX)
      G(1,3)=B*(DEX+GA*(OX-OT))+DB*(OX+NU*OT)-LAM2*D*D1*GA*RAN*(RXE/2.+(
     11.+NU)*OX)
      G(1,4)=LAM2*D1*GA*CX
      G(2,1)=-B*GA*N*(3.-NU)/(2.*RA)-D1*N*DB/(2.*RA)+DNLR*2.*(-1.*(1.+
     1NU)*GA*CTX+GA/8.*(6.*OTX-7.*OX**2-3.*OT**2)-DEX/4.*(5.*CT-3.*CX))
     2-DDNLR/4.*REX*RXE
      G(2,2)=-GA*F(2,2)+D1/2.*B*OTX-B*RAN-LAM2*D*D1*((1.+NU)*CT**2*RAN
     1-CTX/8.*REX**2)
     2   - 2.*MAS/DELSD
      G(2,3)=-B*N*(OT+NU*OX)/RA+DNLR*(GA*DEX-2.*GA2*OX-2.*(1.+NU)*CT
     1*RAN+REX*(GA2+OTX))-DDNLR*REX*GA
      G(2,4)=-NU*LAM2*OT*N/RA
      G(3,1)=-B*GA*(OT+NU*OX)+LAM2*D*D1*(GA*(1.+NU)*(-GA*DEX+GA2*OX
     1-CX*RAN+2.*OTX*CX)+RAN/2.*(GA*OX-GA*OT-3.*DEX))
     2-LAM2*DD*D1*((1.+NU)*GA2*OX+RAN/2.*RXE)
      G(3,2)=-B*N*(OT+NU*CX)/RA+DNLR*(2.*(1.+NU)*(OTX*OT-GA2*OX+2.*GA2
     1*CT-OT*RAN)+GA*DEX+3.*GA2*(CT-OX)+CTX*REX)-DDNLR*(2.*(1.+NU)*GA
     2*CT+GA*REX)
      G(3,3)=-B*(OX**2+2.*NU*OTX+OT**2)+LAM2*D*D1*RAN*((1.+NU)*(OTX-RAN
     1+2.*GA2)+2.*(GA2+CTX))-LAM2*DD*D1*RAN*(3.+NU)*GA
     2   - 2.*MAS/DELSD
      G(3,4)=-LAM2*(D1*CTX+NU*RAN)
      G(4,1)=D*(DEX+NU*GA*OX)
      G(4,2)=D*NU*N*OT/RA
      G(4,3)=D*NU*RAN
      G(4,4)=-1.
      RETURN
      END
      SUBROUTINE POLE(K,P,DEE,DST,X,Z,ZO,Z2,Z3,ZDOT,IS,JS,ID,JD,PHIXB,
     1PHITB)
C***********************************************************************
C         THIS SUBROUTINE PRINTS THE SOLUTION AT AN INITIAL AND A FINAL *
C         POLE.                                                        *
C***********************************************************************
      IMPLICIT LOGICAL*1 ($)
      REAL NU,MT,MX,MTH,MXT,MTS,KX,KT,KXT,LAM,LAM2,MASS
      DIMENSION P(4,4,1),DEE(4,4,1),DST(4,4,1),X(4,1),Z(4,1),ZO(4,1),
     1Z2(4,1),Z3(4,1),ZDOT(4,1),IS(99,1),JS(99,1),ID(99,1),JD(99,1),
     2PHIXB(1),PHITB(1)
      COMMON /IBL2/ N(99),MNINIT
```

```
      COMMON /IBL3/ MO,M1,M2,M3                                         00015120
      COMMON /IBL4/ KMAX,KL                                             00015130
      COMMON /IBL5/ IBCINL,IBCFNL                                       00015140
      COMMON /IBL7/ MNMAXO,MAXD(99),MAXS(99),MAXSY(99),IJS(99)          00015150
      COMMON /IBL8/ LSTEP,ITR                                           00015160
      COMMON /IBL10/ IFREQ,NTHMAX                                       00015170
      COMMON /IBL12/ KMAX1,KMAX2,NCONV                                  00015180
      COMMON /IBLJ/ JUMP                                                00015190
      COMMON /BL4/  ZF1M(4,4,99),ZF2M(4,4,99),
     1              ZF3M(4,4,99),ZF4M(4,4,99)
      COMMON /BL5/  TT(99),MT(99),DT(99),DMT(99)                        00015220
      COMMON /BL6/  SOE,CSE,ALOAD                                       00015230
      COMMON /BL7/  D1,S1                                               00015240
      COMMON /BL8/  R(500),GAM(500),CMT(500)                            00015250
      COMMON /BL10/ PHIX(99),PHIT(99),PHI(99)                           00015260
      COMMON /BL11/ OMXI(500),PHEE,TO,T2                                00015270
      COMMON /BL11A/ PHEN,T2N                                           00015280
      COMMON /BL12/ TDLI,TDEL                                           00015290
      COMMON /BL14/ LAM2,LSD18,LSC1N                                    00015300
      COMMON /BL15/ NU,U1(99),V1(99),W1(99),V2(99),U2(99),W2(99),U3(99),00015310
     1              V3(99),W3(99)                                       00015320
      COMMON /BL17/ DEL                                                 00015330
      COMMON /BL19/ TH(36)                                              00015340
      COMMON /BL20/ DEOMX(500)                                          00015350
      COMMON /BL27/ BX3(99),BT3(99),BXT3(99),BE3(99)                    00015360
      COMMON /BL31/ DELSQ,EXT1(99)                                      00015370
      COMMON /BL32/ TKN,ELAST,CHAR,SIGC                                 00015380
      COMMON /BL100/ TEEC,$DYNMC                                        00015390
      COMMON /BL101/ DELSD                                              00015400
      COMMON /BL102/ DELCAD                                             00015410
      COMMON /BL103/ MASS(500)                                          00015420
      COMMON /BL110/ TX(99),TTH(99),TXT(99),MX(99),MTH(99),MXT(99),     00015430
     1              QS(99)                                              00015440
      COMMON /BL111/ ABZ,ABZO,ABZN,ABZ3,DD2                             00015450
C***********************************************************************00015460
      CALL BCB(K,BS,DB,CS,DD)                                           00015470
      M1L=M1-1                                                          00015480
      M2L=M2-1                                                          00015490
      IF(K.EQ.KMAX) GO TO 301                                           00015500
      DO 202 MN=1,MNMAXO                                                00015510
      U1(MN)=U2(MN)                                                     00015520
      V1(MN)=V2(MN)                                                     00015530
      W1(MN)=W2(MN)                                                     00015540
      I3=3+(MN-1)*KMAX2                                                 00015550
      I2=I3-1                                                           00015560
      U2(MN)=Z(1,I3)                                                    00015570
      V2(MN)=Z(2,I3)                                                    00015580
      W2(MN)=Z(3,I3)                                                    00015590
```

```
      PHIX(MN)=0.
      PHIT(MN)=0.0
      PHI(MN)=0.
      MX(MN)=Z(4,I2)*ABZ3
      MTH(MN)=0.
      MXT(MN)=0.
      CS(MN)=C.
      TX(MN)=0.
      TTH(MN)=0.
 202  TXT(MN)=0.
      IF(M1.EQ.0) GO TO 203
      CALL INLPCL (Z,PHIXB,PHITB)
      PHIX(M1)=PHEE*ABZO
      PHIT(M1)=-PHEE*ABZC
      IF(JUMP.EQ.1) GO TC 1001
       PHIX(M1L)=PHEN*ABZO
      PHIT(M1L)=PHIX(M1L)
 1001 CONTINUE
      II=3+(M1-1)*KMAX2
      IP1=II+1
      IM1=II-1
      GA2=GAM(2)
      CALL BCB(2,BS,DB,CS,DD)
      CALL TLCAD(2,Z)
      PHIXX=-Z(3,IP1)*TDLI+CMXI(2)*Z(1,II)
      PHITT=Z(3,II)/R(2)+CMT(2)*Z(2,II)
      PHII=((Z(2,IP1)-Z(2,IM1))*TCLI+GA2*Z(2,II)+Z(1,II)/R(2))/2.
      CS(M1)=SIGO*TKN*LAM2*((2.-NU)*Z(4,II)-DS*DC2*(PHITT/R(2)+PHIXX
     1   *GA2)+D1*MT(M1)*ALOAD+DS*D1*(-PHIXX/R(2)-GA2*PHITT+(CMT(2)-CMXI
     2      (2))*PHII+(Z(3,IP1)*TDLI-GA2*Z(3,II))/R(2)+GA2*(CMXI(2)-CMT
     3     (2))*Z(2,II)+CMT(2)*(Z(2,IP1)-Z(2,IM1))*TDLI)*.5)/CEL
       IF(MO.EQ.0) GO TO 204
       TX(MO) =TC*ABZ
      TTH(MO) =TO*ABZ
      MTH(MO) =MX(MO)
 204  IF(M2.EQ.0) GO TC 205
       TX(M2) = T2*ABZ
      TTH(M2) =-T2*ABZ
      TXT(M2)=-T2*ABZ
      MTH(M2) =-MX(M2)
      MXT(M2) =MTH(M2)
      IF(JUMP.EQ.1) GO TC 205
      TX(M2L)=T2N*ABZ
      TTH(M2L)=-TX(M2L)
      TXT(M2L)=TX(M2L)
      MTH(M2L)=-MX(M2L)
      MXT(M2L)=MX(M2L)
      GC TO 2C5
```

```
203  IF(MO.EQ.0) GO TO 206
     I3=3+(MO-1)*KMAX2
     I4=I3+1
     CALL TLOAD(1,Z)
     TX(MO)=BS*S1*((2.*Z(1,I3)-.5*Z(1,I4))/DEL+OMXI(1)*Z(3,I3-1))*ABZ
    1        -TT(MO)*ABZ*ALCAC
     TTH(MO)=TX(MO)
     MTH(MO)=MX(MO)
206  IF(M2.EQ.0) GO TO 205
     I3=3+(M2-1)*KMAX2
     I4=I3+1
     TX(M2)=BS*D1*(2.*Z(1,I3)-.5*Z(1,I4))/DEL
      TX(M2) = TX(M2)*ABZ
     TTH(M2) =-TX(M2)
     TXT(M2) =-TX(M2)
     MTH(M2) =-MX(M2)
     MXT(M2) =-MX(M2)
     IF(JUMP.EQ.1) GO TO 205
     TX(M2L)=BS*D1*(2.*Z(1,I3-KMAX2)-.5*Z(1,I4-KMAX2))/DEL
     TX(M2L)=TX(M2L)*ABZ
     TTH(M2L)=-TX(M2L)
     TXT(M2L)=TX(M2L)
     MTH(M2L)=-MX(M2L)
     MXT(M2L)=MX(M2L)
205  RETURN
301  CONTINUE
     DO 302 MN=1,MNMAXC
     U1(MN)=U2(MN)
     V1(MN)=V2(MN)
     W1(MN)=W2(MN)
     PHIX(MN)=0.
     PHIT(MN)=0.
     PHI(MN)=0.
     IK=KMAX1+(MN-1)*KMAX2
     MX(MN)=Z(4,IK)*ABZ3
     MTH(MN)=0.
     MXT(MN)=0.
     CS(MN)=0.
     TX(MN)=0.
     TTH(MN)=0.
302  TXT(MN)=0.
     IF(M1.EQ.0) GO TO 303
     CALL FNLPCL (Z,PHIXB,PHITB)
     PHIX(M1)=PHEE*ABZO
     PHIT(M1)= PHEE*ABZO
     IF(JUMP.EQ.1) GO TO 1002
     PHIX(M1L)=PHEN*ABZO
     PHIT(M1L)=-PHIX(M1L)
```

```
 1002 CONTINUE
      II=KMAX+(M1-1)*KMAX2
      IP1=II+1
      IM1=II-1
      GAK=GAM(KL)
      CALL BCB(KL,BS,DB,CS,DD)
      CALL TLCAD(KL,Z)
      PHIXX=Z(3,IM1)*TDLI+OMXI(KL)*Z(1,II)
      PHITT=Z(3,II)/R(KL)+OMT(KL)*Z(2,II)
      PHII=((Z(2,IP1)-Z(2,IM1))*TDLI+GAK*Z(2,II)+Z(1,II)/R(KL))/2.
      CS(M1)=-SIGO*TKN*LAM2*((2.-NU)*Z(4,II)-DS*DC2*(PHITT/R(KL)+PHIXX
     1*GAK)+C1*MT(M1)*ALCAC-DS*D1*(-PHIXX/R(KL)-GAK*PHITT+(OMT(KL)-OMXI
     2       (KL))*PHII+(-Z(3,IM1)*TDLI-GAK*Z(3,II))/R(KL)+GAK*(OMXI(KL)
     3           -OMT(KL))*Z(2,II)+CMT(KL)*(Z(2,IP1)-Z(2,IM1))*TDLI)*.5)
     4       /CEL
      IF(MO.EQ.0) GO TO 304
       TX(MO) =TO*ABZ
      TTH(MO) =TO*ABZ
      MTH(MO) =MX(MO)
  304 IF(M2.EQ.0) GO TO 305
       TX(M2) = T2*ABZ
      TTH(M2) =-T2*ABZ
      TXT(M2) = T2*ABZ
      MTH(M2) =-MX(M2)
      MXT(M2) =MX(M2)
      IF(JUMP.EQ.1) GO TO 305
      TX(M2L)=T2N*ABZ
      TTH(M2L)=-TX(M2L)
      TXT(M2L)=-TX(M2L)
      MTH(M2L)=-MX(M2L)
      MXT(M2L)=-MX(M2L)
      GO TO 305
  303 IF(MO.EQ.0) GO TO 306
      IKM=KMAX+(MO-1)*KMAX2
      IM1=IKM-1
      CALL TLCAD(KMAX,Z)
      TX(MO)=BS*S1*((-2.*Z(1,IKM)+.5*Z(1,IM1))/CEL+CMXI(KMAX)*Z(3,IKM+1)
     1          )*ABZ-TT(MO)*ABZ*ALCAD
      TTH(MO) =TX(MO)
      MTH(MO) =MX(MO)
  306 IF(M2.EQ.0) GO TO 305
      IKM=KMAX+(M2-1)*KMAX2
      IM1=IKM-1
      TX(M2)=BS*D1*(-2.*Z(1,IKM)+.5*Z(1,IM1))/CEL
      TX(M2)=TX(M2)*ABZ
      TTH(M2) =-TX(M2)
      TXT(M2) = TX(M2)
      MTH(M2) =-MX(M2)
```

```
      MXT(M2) = MX(M2)
      IF(JUMP.EQ.1) GO TO 305
      TX(M2L)=BS*D1*(-2.*Z(1,IKM-KMAX2)+.5*Z(1,IM1-KMAX2))/DEL
      TX(M2L)=TX(M2L)*ABZ
      TTH(M2L)=-TX(M2L)
      TXT(M2L)=-TX(M2L)
      MTH(M2L)=-MX(M2L)
      MXT(M2L)=-MX(M2L)
  305 RETURN
      END
      SUBROUTINE MODES (IS,JS,ID,JD,P,X,ZO,Z2,Z3,DEE,DST)
C*********************************************************************
C          IN THIS SUBROUTINE, ARRAYS THAT DEFINE THOSE SETS OF INDICES  *
C          THAT COMBINE TO EQUAL EACH VALUE OF N IN THE PROBLEM ARE DETER*
C          MINED.  MODES IS CALLED PRIOR TO THE FIRST ITERATION AND AFTER*
C          EVERY ITERATION UNTIL A SPECIFIED NUMBER OF FOURIER TERMS IS  *
C          REACHED.  EACH FOURIER INDEX IN THE PROBLEM IS SUBTRACTED FROM*
C          ALL OTHER FOURIER INDICES AND THE RESULT IS COMPARED WITH ALL *
C          FOURIER INDICES TO SEE IF THE NEW VALUE EXISTS IN THE PROGRAM.*
C          IF IT DOES, THE LOCATIONS OF THE TWO INDICES THAT MADE THE COM-
C          BINATION ARE STORED IN TWO SPECIAL 2-DIMENSIONAL ARRAYS, ID AND
C          JD.  ONE ARGUMENT OF EACH ARRAY IS THE VALUE OF THE NEW INDEX *
C          AND THE OTHER IS THE NUMBER OF COMBINATIONS OF INDICES THAT   *
C          ALSO GIVE THIS VALUE OF THE INDEX.  IF THERE IS NO INDEX IN THE
C          PROGRAM THAT MATCHES THE NEW ONE, THEN A NEW FOURIER TERM HAS *
C          BEEN GENERATED AND WILL BE CONSIDERED IN THE NEXT ITERATION FOR
C          SOLUTION.                                                     *
C          THE VARIABLE MAXD STORES THE TOTAL NUMBER OF SUCH COMBINATIONS*
C          FOR EACH VALUE OF THE FOURIER INDEX.  IN A SIMILAR MANNER, EACH
C          INDEX IS ADDED TO EVERY OTHER INDEX AND THE SUM COMPARED WITH *
C          ALL INDICES.  THIS RESULT IS STORED IN THE 2-DIMENSIONAL ARRAYS
C          IS AND JS, IN THE SAME MANNER AS WAS DONE FOR THE SUBTRACTION *
C          CASE.  THE VARIABLE MAXS STORES THE TOTAL NUMBER OF SUMMATION *
C          COMBINATIONS FOR EACH VALUE OF THE FOURIER INDEX.             *
C          A SPECIAL ROUTINE HANDLES THE CASES WHERE THE INDEX IS ADDED TO
C          AND SUBTRACTED FROM ITSELF.  THE 2-DIMENSIONAL ARRAY IJS STORES
C          THE LOCATION OF THE INDEX AND THE VARIABLE MAXSY STORES THE   *
C          TOTAL NUMBER OF SUCH COMBINATIONS.                            *
C          WITH THIS PROCEDURE, THE SERIES OF PRODUCTS THAT MAKE UP THE  *
C          BETA'S AND AETA'S CONTAIN NO ZERO TERM, AND THE SUMMATION IS  *
C          CARRIED OUT IN PHIBET(K) AND TEAETA(K) OVER SPECIFICALLY DE-  *
C          FINED LIMITS.                                                 *
C*********************************************************************
      DIMENSION P(4,4,1),DEE(4,4,1),DST(4,4,1),X(4,1),ZO(4,1),Z2(4,1),
     1Z3(4,1),IS(99,1),JS(99,1),ID(99,1),JD(99,1)
      COMMON /IBL1/ MNMAX
      COMMON /IBL2/ N(99),MNINIT
      COMMON /IBL7/ MNMAXC,MAXD(99),MAXS(99),MAXSY(99),IJS(99)
```

```
      COMMON /IBL9/ MAXM
      COMMON /IBL11/ ICORFL,IPASS
     7/IELJ/ JUMP
C***********************************************************************
C     DETERMINE IF NEED FOR MODAL COUPLING EXISTS                      *
C***********************************************************************
      IF(MAXM.EQ.1) RETURN
C***********************************************************************
C     SEE IF ENOUGH NEW MODES HAVE ALREADY BEEN GENERATED              *
C***********************************************************************
      IF(MNINIT.GT.MAXM) RETURN
C***********************************************************************
C                                                                      *
C                                                                      *
C     THIS SECTION HANDLES MODES GENERATED FROM THE DIFFERENCE         *
C     COMBINATIONS OF EXISTING MODES                                   *
C                                                                      *
C                                                                      *
C***********************************************************************
C     SET UP INITIAL LOOP FOR DIFFERENCE COMBINATIONS                  *
C***********************************************************************
      DO 1 MN=1,MNMAXO,JUMP
      NMN=N(MN)
      NNS=MN
      IF(MNINIT.GT.MN) NNS=MNINIT
C*******************************************Q*******QQQ*Q4**********
C     SET UP SECOND LOOP - TO SUBTRACT PRESENT MODE FROM ALL OTHER     *
C     MODES                                                            *
C***********************************************************************
      DO 1 MM=NNS,MNMAXO,JUMP
      NMM=N(MM)
C***********************************************************************
C     CREATE DIFFERENCE                                                *
C***********************************************************************
      NTEST=IABS(NMN-NMM)
C***********************************************************************
C     SET UP 3RD LOOP - TO COMPARE DIFFERENCE WITH ALL OTHER MODES     *
C***********************************************************************
      DO 2 MMFT=1,MNMAX,JUMP
C***********************************************************************
C     IF WE SATISFY HERE, MODE EXISTS, GO TO INCREMENT -MAXO-          *
C***********************************************************************
      IF(NTEST.EQ.N(MMFT)) GO TO 10
    2 CONTINUE
C***********************************************************************
C     IF WE MAKE IT TO HERE, WE HAVE GENERATED A NEW MODE              *
C***********************************************************************
C     DO WE WANT ANY MORE NEW MODES                                    *
```

```
C***********************************************************************
      IF(ICORFL.EQ.1) GO TO 1
C***********************************************************************
C     FROM HERE TO -1-, WE INCREMENT COUNTERS AND MAKE APPROPRIATE     *
C     ENTRIES IN -MAXD-,-IC- AND -JC-. TO SIMPLIFY USAGE OF -IC-       *
C     AND -JC- IN TEAETA AND PHIBET, MODES ARE PLACED IN THESE         *
C     ARRAYS SUCH THAT THE HIGHER MODE NUMBER IF ALWAYS FIRST          *
C***********************************************************************
      MNMAX=MNMAX+JUMP
      N(MNMAX)=NTEST
      IF (JUMP.GT.1) N(MNMAX-1)=-NTEST
      MMFT=MNMAX
      IF(MNMAX.EQ.MAXM) ICORFL=1
   10 IF(NMN-NMM) 11,1,12
   11 LOCD=MAXD(MMFT)+1
      MAXD(MMFT)=LOCD
      IC(LOCD,MMFT)=MM
      JC(LOCD,MMFT)=MN
      GO TO 1
   12 LOCD=MAXD(MMFT)+1
      MAXD(MMFT)=LOCD
      IC(LOCD,MMFT)=MN
      JC(LOCD,MMFT)=MM
    1 CONTINUE
C***********************************************************************
C                                                                      *
C                                                                      *
C     THIS SECTION HANDLES MODES GENERATED FROM THE SUMMATION          *
C     COMBINATIONS OF EXISTING MODES                                   *
C                                                                      *
C                                                                      *
C***********************************************************************
C     NOW SET UP INITIAL LOOP FOR SUM COMBINATIONS                     *
C***********************************************************************
      DO 301 MN=1,MNMAXO,JUMP
      NMN=N(MN)
      NNS=MN
      IF(MNINIT.GT.MN) NNS=MNINIT
C***********************************************************************
C     SET UP SECOND LOOP - TO ADD PRESENT MODE TO ALL OTHER MODES      *
C***********************************************************************
      DO 301 MM=NNS,MNMAXO,JUMP
      NMM=N(MM)
C***********************************************************************
C     CREATE SUM                                                       *
C***********************************************************************
      NTEST=NMN+NMM
C***********************************************************************
```

```
C      SET UP THIRD LOOP - TO COMPARE SUM WITH ALL OTHER MODES              *
C***********************************************************************
      DO 302 MMFT=1,MNMAX,JUMP
C***********************************************************************
C      IF WE SATISFY HERE, MODE EXISTS, GO TO INCREMENT -MAXS- OR           *
C      -MAXSY-)                                                             *
C***********************************************************************
      IF(NTEST.EQ.N(MMFT)) GO TO 310
  302 CONTINUE
C***********************************************************************
C      IF WE MAKE IT TO HERE, WE HAVE GENERATED A NEW MODE                  *
C***********************************************************************
C      DO WE WANT ANY MORE NEW MODES                                         *
C***********************************************************************
      IF(ICORFL.EQ.1) GO TO 301
      IF(MNMAX.GE.MAXM) GO TO 301
C***********************************************************************
C      INCREMENT -MNMAX- AND ESTABLISH NEW MODE NUMBER                      *
C***********************************************************************
      MNMAX=MNMAX+JUMP
      N(MNMAX)=NTEST
      IF (JUMP.GT.1)  N(MNMAX-1)=-NTEST
      MMFT=MNMAX
      IF(MNMAX.GE.MAXM) ICORFL=1
C***********************************************************************
C      IF MODE WAS ADDED TO ITSELF, GO TO -MAXSY AND IJS- SECTION           *
C***********************************************************************
  310 IF(NMN.EQ.NMM) GO TO 360
C***********************************************************************
C      MAKE ENTRIES IN -LOCS-,-IS- AND -JS-                                 *
C***********************************************************************
      LOCS=MAXS(MMFT)+1
      MAXS(MMFT)=LOCS
      IS(LOCS,MMFT)=MN
      JS(LOCS,MMFT)=MM
      GO TO 301
C***********************************************************************
C      SEE IF THE SUM OF THE MODE WITH ITSELF WAS THE 0-TH MODE             *
C***********************************************************************
  360 IF(NMN.EQ.0) GO TO 301
C***********************************************************************
C      IF HERE, IT WASN-T, MAKE ENTRIES IN -MAXSY- AND -IJS-                *
C***********************************************************************
      MAXSY(MMFT)=1
      IJS(MMFT)=MN
  301 CONTINUE
      MNINIT=MNMAXO+JUMP
      IF(ICORFL.GT.0) IPASS=IPASS+1
```

```
      IF(IPASS.LT.2.AND.MNINIT.LE.MNMAX) CALL PMATRX (P,X,ZC,Z2,Z3,CEE,
     1CST)
      RETURN
      END
      SUBROUTINE TEAETA(K,Z,IS,JS,ID,JC)
C***********************************************************************
C         THIS SUBROUTINE CALCULATES THE INPLANE FORCES AND CARRIES OUT *
C         THE MULTIPLYING AND SUMMATION PROCEDURE FOR COMPUTINE THE ETA *
C         NON-LINEAR TERMS FOR A GIVEN MERIDIONAL STATION K.  THE ARRAYS*
C         IS, JS, ID, JS, IJS, MAXS, MAXD, MAXSY PREPARED IN SUBROUTINE *
C         MODES ARE USED HERE                                           *
C***********************************************************************
      REAL NU,MT
      DIMENSION Z(4,1),IS(99,1),JS(99,1),ID(99,1),JC(99,1)
      COMMON /IBL1/ MNMAX
      COMMON /IBL2/ N(99),MNINIT
      COMMON /IBL7/ MNMAXO,MAXD(99),MAXS(99),MAXSY(99),IJS(99)
      COMMON /IBL8/ LSTEP,ITR
      COMMON /IBL13/ ITRMAX,LSMAX
     5/IBLJ/ JUMP
      COMMON /BL5/  TT(99),MT(99),DT(99),CMT(99)
      COMMON /BL6/  SOE,CSE,ALOAD
      COMMON /BL7/  D1,S1
      COMMON /BL8/  R(500),GAM(500),OMT(500)
      COMMON /BL10/ PHIX(99),PHIT(99),PHI(99)
      COMMON /BL11/ OMXI(500),PHEE,T0,T2
      COMMON /BL12/ TDLI,TDEL
      COMMON /BL15/ NU,U1(99),V1(99),W1(99),V2(99),U2(99),W2(99),U3(99),
     1              V3(99),W3(99)
      COMMON /BL27/ BX3(99),BT3(99),BXT3(99),BE3(99)
      COMMON /BL28/ EXX3(99),ETT3(99),ETX3(99),EXT3(99),EX3(99),ET3(99)
      COMMON /BLPHS/ PHX(99),PHT(99)
      DIMENSION TX(99),TTH(99),TXT(99)
C***********************************************************************
      RRA=1./R(K)
      GA=GAM(K)
      OX=OMXI(K)
      OT=OMT(K)
      CALL BDE(K,BS,DB,CS,DD)
      DO 1 M=1,MNMAXO
      PHIX(M)=PHIX(M)+PHX(M)
      PHIT(M)=PHIT(M)+PHT(M)
      EN=N(M)
      CALL TLOAD(K,Z)
      TTS=TT(M)*ALOAD
      EX=(U3(M)-U1(M))*TDLI+OX*W2(M)+CSE*(BX3(M)+BE3(M))
      ET= EN *V2(M)*RRA+GA*U2(M)+OT*W2(M)+OSE*(BT3(M)+BE3(M))
      EXT=.5*(TDLI*(V3(M)-V1(M))- EN *U2(M)*RRA-GA*V2(M))+CSE*EXT3(M)
```

```
      TX(M)=BS*(EX+NU*ET)-TTS
      TTH(M)=BS*(ET+NU*EX)-TTS
    1 TXT(M)=BS*C1*EXT
      IF(JUMP.EQ.2) GO TO 1111
      DO 9 M=1,MNMAX
      SMF=0.
      SMS=0.
      SMV=0.
      SME=0.
      SMN=0.
      SMT=0.
      IF(N(M).EQ.0) GO TO 20
      MAXL=MAXS(M)
      IF(MAXL.EQ.0) GO TO 2
      DO 3 L=1,MAXL
      I=IS(L,M)
      J=JS(L,M)
      SMF=SMF+TX(I)*PHIX(J)+TX(J)*PHIX(I)
      SMS=SMS+TTH(I)*PHIT(J)+TTH(J)*PHIT(I)
      SMV=SMV-PHIT(I)*TXT(J)-PHIT(J)*TXT(I)
      SME=SME+PHIX(I)*TXT(J)+PHIX(J)*TXT(I)
      SMN=SMN+TX(I)*PHI(J)+TX(J)*PHI(I)
    3 SMT=SMT+TTH(I)*PHI(J)+TTH(J)*PHI(I)
    2 MAXL=MAXD(M)
      IF(MAXL.EQ.0) GO TO 4
      DO 5 L=1,MAXL
      I=ID(L,M)
      J=JD(L,M)
      SMF=SMF+TX(I)*PHIX(J)+TX(J)*PHIX(I)
      SMS=SMS-TTH(I)*PHIT(J)+TTH(J)*PHIT(I)
      SMV=SMV+PHIT(I)*TXT(J)+PHIT(J)*TXT(I)
      SME=SME-PHIX(I)*TXT(J)+PHIX(J)*TXT(I)
      SMN=SMN-TX(I)*PHI(J)+TX(J)*PHI(I)
    5 SMT=SMT-TTH(I)*PHI(J)+TTH(J)*PHI(I)
    4 IF(MAXSY(M).EQ.0) GO TO 10
      I=IJS(M)
      SMF=SMF+TX(I)*PHIX(I)
      SMS=SMS+TTH(I)*PHIT(I)
      SMV=SMV-PHIT(I)*TXT(I)
      SME=SME+PHIX(I)*TXT(I)
      SMN=SMN+TX(I)*PHI(I)
      SMT=SMT+TTH(I)*PHI(I)
      GO TO 10
   20 DO 21 L=1,MNMAXO
      SMF=SMF+TX(L)*PHIX(L)
   21 SMV=SMV+PHIT(L)*TXT(L)
      IF(M.GT.MNMAXO) GO TO 10
      SMF=SMF+TX(M)*PHIX(M)
```

```
   10 EXX3(M)=SMF*.5
      ETT3(M)=SMS*.5
      ETX3(M)=SMV*.5
      EXT3(M)=SME*.5
      EX3(M)=SMN*.5
      ET3(M)=SMT*.5
    9 CONTINUE
      GO TO 2222
C***********************************************************************
C                                                                      *
C                                                                      *
C     THIS SECTION HANDLES GENERAL LOADING                             *
C                                                                      *
C                                                                      *
C***********************************************************************
 1111 DO 49 M=1,MNMAX
      SMF=0.0
      SMS=0.0
      SMV=0.0
      SME=0.0
      SMN=0.0
      SMT=0.0
C***********************************************************************
C     TEST FOR ASYMMETRIC MODE                                         *
C***********************************************************************
      IF (N(M).LT.0) GO TO 101
C***********************************************************************
C                                                                      *
C     THIS SECTION HANDLES SYMMETRIC COMBINATIONS                      *
C                                                                      *
C***********************************************************************
C     TEST FOR ZERO-TH MODE                                            *
C***********************************************************************
      IF (N(M).EQ.0) GO TO 420
      MAXL=MAXS(M)
C***********************************************************************
C     TEST FOR PRESENCE OF SYMMETRIC SUMMATION COMBINATIONS            *
C***********************************************************************
      IF (MAXL.EQ.0) GO TO 42
C***********************************************************************
C     SET UP COUPLING MODES- INDICES AND TEST FOR MODE 1               *
C***********************************************************************
      DO 43 L=1,MAXL
      I=IS(L,M)
      J=JS(L,M)
      II=I-1
      JJ=J-1
      IF (I.EQ.1) GO TO 43
```

```
C***********************************************************************
C     COMPILE SUMS FOR SYMMETRIC SUM COMBINATION MODES                 *
C***********************************************************************
      SMF=SMF+TX(I)*PHIX(J)+TX(J)*PHIX(I)-PHIX(II)*TX(JJ)
     1         -PHIX(JJ)*TX(II)
      SMS=SMS+TTH(I)*PHIT(J)+TTH(J)*PHIT(I)+PHIT(II)*TTH(JJ)
     1         +PHIT(JJ)*TTH(II)
      SMV=SMV-PHIT(I)*TXT(J)-PHIT(J)*TXT(I)+PHIT(II)*TXT(JJ)
     1         +PHIT(JJ)*TXT(II)
      SME=SME+PHIX(I)*TXT(J)+PHIX(J)*TXT(I)+PHIX(II)*TXT(JJ)
     1         +PHIX(JJ)*TXT(II)
      SMN=SMN+TX(I)*PHI(J)+TX(J)*PHI(I)+PHI(II)*TX(JJ)
     1         +PHI(JJ)*TX(II)
      SMT=SMT+TTH(I)*PHI(J)+TTH(J)*PHI(I)+PHI(II)*TTH(JJ)
     1         +PHI(JJ)*TTH(II)
   43 CONTINUE
   42 MAXL=MAXD(M)
C***********************************************************************
C     TEST FOR PRESENCE OF SYMMETRIC DIFFERENCE COMBINATIONS           *
C***********************************************************************
      IF (MAXL.EQ.0) GO TO 44
C***********************************************************************
C     SET UP COUPLING MODES- INDICES AND TEST FOR MODE 1               *
C***********************************************************************
      DO 45 L=1,MAXL
      I=ID(L,M)
      J=JD(L,M)
      II=I-1
      JJ=J-1
      IF (J.EQ.1) GO TO 442
C***********************************************************************
C     COMPILE SUMS FOR SYMMETRIC DIFFERENCE COMBINATIONS
C***********************************************************************
      SMF=SMF+TX(I)*PHIX(J)+TX(J)*PHIX(I)+PHIX(II)*TX(JJ)
     1         +PHIX(JJ)*TX(II)
      SMS=SMS-TTH(I)*PHIT(J)+TTH(J)*PHIT(I)-PHIT(II)*TTH(JJ)
     1         +PHIT(JJ)*TTH(II)
      SMV=SMV+PHIT(I)*TXT(J)+PHIT(J)*TXT(I)+PHIT(II)*TXT(JJ)
     1         +PHIT(JJ)*TXT(II)
      SME=SME-PHIX(I)*TXT(J)+PHIX(J)*TXT(I)+PHIX(II)*TXT(JJ)
     1         -PHIX(JJ)*TXT(II)
      SMN=SMN-TX(I)*PHI(J)+TX(J)*PHI(I)-PHI(II)*TX(JJ)
     1         +PHI(JJ)*TX(II)
      SMT=SMT-TTH(I)*PHI(J)+TTH(J)*PHI(I)-PHI(II)*TTH(JJ)
     1         +PHI(JJ)*TTH(II)
      GO TO 45
C***********************************************************************
C     EXECUTE BELOW IF J=1 IN DIFF-COMB. (OR I=1 IN SUM COMB)          *
```

```
C***********************************************************************
  442 SMF=SMF+(PHIX(1)*TX(I)+PHIX(I)*TX(1))*2.0
      SMS=SMS+(PHIT(I)*TTH(1)+PHIT(1)*TTH(II))*2.0
      SMV=SMV+(PHIT(1)*TXT(II)+PHIT(II)*TXT(1))*2.0
      SME=SME+(PHIX(II)*TXT(1)+PHIX(1)*TXT(I))*2.0
      SMN=SMN+(PHI(I)*TX(1)+PHI(1)*TX(II))*2.0
      SMT=SMT+(PHI(I)*TTH(1)+PHI(1)*TTH(II))*2.0
   45 CONTINUE
C***********************************************************************
C      TEST FOR PRESENCE OF SAME-INDEX COMBINATIONS                    *
C***********************************************************************
   44 IF (MAXSY(M).EQ.0) GO TO 410
C***********************************************************************
C      SET UP INDICES AND COMPILE SUMS                                 *
C***********************************************************************
      I=IJS(M)
      II=I-1
      SMF=SMF+TX(I)*PHIX(I)-PHIX(II)*TX(II)
      SMS=SMS+TTH(I)*PHIT(I)+PHIT(II)*TTH(II)
      SMV=SMV-PHIT(I)*TXT(I)+PHIT(II)*TXT(II)
      SME=SME+PHIX(I)*TXT(I)+PHIX(II)*TXT(II)
      SMN=SMN+TX(I)*PHI(I)+PHI(II)*TX(II)
      SMT=SMT+TTH(I)*PHI(I)+PHI(II)*TTH(II)
      GO TO 410
C***********************************************************************
C      HERE WE HANDLE CASES WHERE N(M)=0                               *
C***********************************************************************
  420 DO 421 L=1,MNMAXO
      SMF=SMF+TX(L)*PHIX(L)
      SMV=SMV+TXT(L)*PHIT(L)
  421 CONTINUE
      DO 422 L=3,MNMAXO,2
      LL=L-1
      SMS=SMS+PHIT(LL)*TTH(L)+PHIT(L)*TTH(LL)
      SME=SME+PHIX(L)*TXT(LL)+PHIX(LL)*TXT(L)
      SMN=SMN+PHI(LL)*TX(L)+PHI(L)*TX(LL)
      SMT=SMT+PHI(LL)*TTH(L)+PHI(L)*TTH(LL)
  422 CONTINUE
      SMF=SMF+TX(1)*PHIX(1)
      SMS=SMS+TTH(1)*PHIT(1)*2.0
      SMV=SMV+TXT(1)*PHIT(1)
      SME=SME+TXT(1)*PHIX(1)*2.0
      SMN=SMN+TX(1)*PHI(1)*2.0
      SMT=SMT+TTH(1)*PHI(1)*2.0
      GO TO 410
C***********************************************************************
C                                                                      *
C      THIS SECTION HANDLES ASYMMETRIC MODE COMBINATIONS               *
```

```
C
C***********************************************************************
  101 MP=M+1
      MAXL=MAXS(MP)
C***********************************************************************
C     TEST FOR PRESENCE OF SUMMATION COMBINATIONS                      *
C***********************************************************************
      IF (MAXL.EQ.0) GO TO 102
C***********************************************************************
C     SET UP COUPLING INDICES AND TEST FOR MODE 1                      *
C***********************************************************************
      DO 103 L=1,MAXL
      I=IS(L,MP)
      J=JS(L,MP)
      II=I-1
      JJ=J-1
      IF (I.EQ.1) GO TO 103
C***********************************************************************
C     COMPILE SUMS FOR ASYMMETRIC SUMMATION COMBINATIONS               *
C***********************************************************************
      SMF=SMF+PHIX(I)*TX(JJ)+PHIX(J)*TX(II)+PHIX(II)*TX(J)
     1        +PHIX(JJ)*TX(I)
      SMS=SMS-PHIT(I)*TTH(JJ)-PHIT(J)*TTH(II)+PHIT(II)*TTH(J)
     1        +PHIT(JJ)*TTH(I)
      SMV=SMV+PHIT(I)*TXT(JJ)+PHIT(J)*TXT(II)+PHIT(II)*TXT(J)
     1        +PHIT(JJ)*TXT(I)
      SME=SME+PHIX(I)*TXT(JJ)+PHIX(J)*TXT(II)-PHIX(II)*TXT(J)
     1        -PHIX(JJ)*TXT(I)
      SMN=SMN-PHI(I)*TX(JJ)-PHI(J)*TX(II)+PHI(II)*TX(J)
     1        +PHI(JJ)*TX(I)
      SMT=SMT-PHI(I)*TTH(JJ)-PHI(J)*TTH(II)+PHI(II)*TTH(J)
     1        +PHI(JJ)*TTH(I)
  103 CONTINUE
C***********************************************************************
C     TEST FOR PRESENCE OF DIFFERENCE COMBINATIONS                     *
C***********************************************************************
  102 MAXL=MAXD(MP)
      IF (MAXL.EQ.0) GO TO 104
C***********************************************************************
C     SET UP COUPLING MODES- INDICES AND TEST FOR MODE 1               *
C***********************************************************************
      DO 105 L=1,MAXL
      I=ID(L,MP)
      J=JD(L,MP)
      II=I-1
      JJ=J-1
      IF (J.EQ.1) GO TO 123
C***********************************************************************
```

```
C       COMPILE SUMS FOR ASYMMETRIC DIFFERENCE COMBINATIONS                  *
C*****************************************************************************
        SMF=SMF-PHIX(I)*TX(JJ)+PHIX(J)*TX(II)+PHIX(II)*TX(J)
     1        -PHIX(JJ)*TX(I)
        SMS=SMS+PHIT(I)*TTH(JJ)+PHIT(J)*TTH(II)+PHIT(II)*TTH(J)
     1        +PHIT(JJ)*TTH(I)
        SMV=SMV+PHIT(I)*TXT(JJ)+PHIT(J)*TXT(II)+PHIT(II)*TXT(J)
     1        +PHIT(JJ)*TXT(I)
        SME=SME+PHIX(I)*TXT(JJ)+PHIX(J)*TXT(II)+PHIX(II)*TXT(J)
     1        +PHIX(JJ)*TXT(I)
        SMN=SMN+PHI(I)*TX(JJ)+PHI(J)*TX(II)+PHI(II)*TX(J)
     1        +PHI(JJ)*TX(I)
        SMT=SMT+PHI(I)*TTH(JJ)+PHI(J)*TTH(II)+PHI(II)*TTH(J)
     1        +PHI(JJ)*TTH(I)
        GO TO 105
C*****************************************************************************
C       EXECUTE BELOW IF J=1 IN DIFF-COMB (OR I=1 IN SUM-COMB)               *
C*****************************************************************************
  123   SMF=SMF+(PHIX(1)*TX(II)+PHIX(II)*TX(1))*2.0
        SMS=SMS+(PHIT(1)*TTH(I)+PHIT(II)*TTH(1))*2.0
        SMV=SMV+(PHIT(I)*TXT(1)+PHIT(1)*TXT(I))*2.0
        SME=SME+(PHIX(1)*TXT(II)+PHIX(I)*TXT(1))*2.0
        SMN=SMN+(PHI(1)*TX(I)+PHI(II)*TX(1))*2.0
        SMT=SMT+(PHI(1)*TTH(I)+PHI(II)*TTH(1))*2.0
  105   CONTINUE
C*****************************************************************************
C       TEXT FOR PRESENCE OF SAME-INDEX COMBINATION                          *
C*****************************************************************************
  104   IF (MAXSY(MP).EQ.0) GO TO 410
C*****************************************************************************
C       SET UP COUPLING MODES- INDICES AND COMPILE SUMS                      *
C*****************************************************************************
        I=IJS(MP)
        II=I-1
        SMF=SMF+PHIX(I)*TX(II)+PHIX(II)*TX(I)
        SMS=SMS-PHIT(I)*TTH(II)+PHIT(II)*TTH(I)
        SMV=SMV+PHIT(I)*TXT(II)+PHIT(II)*TXT(I)
        SME=SME+PHIX(I)*TXT(II)-PHIX(II)*TXT(I)
        SMN=SMN-PHI(I)*TX(II)+PHI(II)*TX(I)
        SMT=SMT-PHI(I)*TTH(II)+PHI(II)*TTH(I)
  410   CONTINUE
C*****************************************************************************
C       PREPARE -AETA- TERMS                                                 *
C*****************************************************************************
        EXX3(M)=SMF*0.5
        ETT3(M)=SMS*0.5
        ETX3(M)=SMV*0.5
        EXT3(M)=SME*0.5
```

```
      EX3(M) =SMA*0.5                                                   00022320
      ET3(M) =SMT*0.5                                                   00022330
   49 CONTINUE                                                          00022340
 2222 CONTINUE                                                          00022350
C***********************************************************************00022360
C     UP-DATE U,V,W AND RETURN                                         *00022370
C***********************************************************************00022380
      DO 30 M=1,MNMAXO                                                  00022390
      U1(M)=U2(M)                                                       00022400
      V1(M)=V2(M)                                                       00022410
      W1(M)=W2(M)                                                       00022420
      U2(M)=U3(M)                                                       00022430
      V2(M)=V3(M)                                                       00022440
   30 W2(M)=W3(M)                                                       00022450
      RETURN                                                            00022460
      END                                                               00022470
      BLOCK DATA
      COMMON /BL27/ BX3    ,BT3    ,BXT3   ,BE3
      COMMON /BL28/ EXX3    ,ETT3    ,ETX3    ,EXT3    ,EX3    ,ET3
      COMMON /BL29/ BX1    ,BT1    ,BXT1   ,BE1    ,BX2    ,BT2    ,
     1              BXT2   ,BE2
      COMMON /BL30/ EXX1    ,ETT1    ,ETX1    ,EX1    ,ET1    ,EXX2    ,
     1              ETT2    ,ETX2    ,EXT2    ,EX2    ,ET2
      COMMON /BL31/ DELSQ,EXT1
      REAL BX3(99)/99*0./,BT3(99)/99*0./,BXT3(99)/99*0./,BE3(99)/99*0./,
     1EXX3(99)/99*0./,ETT3(99)/99*0./,ETX3(99)/99*0./,EXT3(99)/99*0./,
     2EX3(99)/99*0./,ET3(99)/99*0./,BX1(99)/99*0./,BT1(99)/99*0./,
     3BXT1(99)/99*0./,BE1(99)/99*0./,BX2(99)/99*0./,BT2(99)/99*0./,
     4BXT2(99)/99*0./,BE2(99)/99*0./,EXX1(99)/99*0./,ETT1(99)/99*0./,
     5ETX1(99)/99*0./,EX1(99)/99*0./,ET1(99)/99*0./,EXX2(99)/99*0./,
     6ETT2(99)/99*0./,ETX2(99)/99*0./,EXT2(99)/99*0./,EX2(99)/99*0./,
     7ET2(99)/99*0./,DELSQ/1*0./,EXT1(99)/99*0./
      END
//GO.SYSIN DD *

DATA CARDS GO HERE
```

APPENDIX B

LISTING OF OUTPUT FROM EXAMPLE PROBLEM

```
                                        --PROBLEM NUMBER     1--

                                  LMSC TEST CASE, IMPULSIVELY LOADED CONE

                                         --INPUT DATA RECORD--
     THE BOUNDARY CONDITIONS ARE:

         AT THE INITIAL EDGE
 ---------------OMEGA BAR---------------            ---------------LAMDA BAR---------------          ---FI-----
 ( 0.0      0.0      0.0      0.0      )  NS        ( 0.100E 01 0.0       0.0       0.0   )  U        0.0
 ( 0.0      0.0      0.0      0.0      )  NST   +   ( 0.0       0.100E 01 0.0       0.0   )  V   =    0.0
 ( 0.0      0.0      0.0      0.0      )  QS        ( 0.0       0.0       0.100E 01 0.0   )  W        0.0
 ( 0.0      0.0      0.0      0.100E 01)  PHIS      ( 0.0       0.0       0.0       0.0   )  MS       0.0
         AT THE FINAL EDGE
 ---------------OMEGA BAR---------------            ---------------LAMDA BAR---------------          ---FI-----
 ( 0.0      0.0      0.0      0.0      )  NS        ( 0.100E 01 0.0       0.0       0.0   )  U        0.0
 ( 0.0      0.0      0.0      0.0      )  NST   +   ( 0.0       0.100E 01 0.0       0.0   )  V   =    0.0
 ( 0.0      0.0      0.0      0.0      )  QS        ( 0.0       0.0       0.100E 01 0.0   )  W        0.0
 ( 0.0      0.0      0.0      0.100E 01)  PHIS      ( 0.0       0.0       0.0       0.0   )  MS       0.0


     NUMBER OF STATIONS---------------------- 31
     NUMBER OF MODES------------------------- 4
     INCREMENTAL TIME------------------------.1824E-01
     MAXIMUM NUMBER OF TIME STEPS------------ 750
     MAXIMUM NUMBER OF ITERATIONS------------ 20
     CONVERGENCE CRITERION-------------------0.0100

     CHARACTERISTIC SHELL DIMENSION----------  0.1500E 02
     REFERENCE THICKNESS---------------------  0.5430E 00
     REFERENCE ELASTICITY--------------------  0.3520E 07
     REFERENCE STRESS------------------------  0.1000E 04
     REFERENCE TIME--------------------------  0.1095E-03
     POISSONS RATIO----------------------  0.2860E 00

               CIRCUMFERENTIAL COORDINATES FOR THE PRINT RECORD, IN RADIAN MEASURE, ARE:
          0.0             3.1415920258

               THE DATA PRINTED IS DIMENSIONAL
         EXECUTING IN SUBROUTINE "DYNAMIC"
```

STATION	RADIUS	GAMMA	OMEGA S	OMEGA THETA	DEOMEGA S	MASS
1	0.7950E 01	0.1912E-01	0.0	0.1243E 00	0.0	0.1021E-03
2	0.8026E 01	0.1894E-01	0.0	0.1231E 00	0.0	0.1021E-03
3	0.8102E 01	0.1876E-01	0.0	0.1220E 00	0.0	0.1021E-03
4	0.8178E 01	0.1858E-01	0.0	0.1209E 00	0.0	0.1021E-03
5	0.8254E 01	0.1841E-01	0.0	0.1197E 00	0.0	0.1021E-03
6	0.8330E 01	0.1824E-01	0.0	0.1187E 00	0.0	0.1021E-03
7	0.8406E 01	0.1808E-01	0.0	0.1176E 00	0.0	0.1021E-03
8	0.8482E 01	0.1792E-01	0.0	0.1165E 00	0.0	0.1021E-03
9	0.8558E 01	0.1776E-01	0.0	0.1155E 00	0.0	0.1021E-03
10	0.8634E 01	0.1760E-01	0.0	0.1145E 00	0.0	0.1021E-03
11	0.8710E 01	0.1745E-01	0.0	0.1135E 00	0.0	0.1021E-03
12	0.8786E 01	0.1730E-01	0.0	0.1125E 00	0.0	0.1021E-03
13	0.8862E 01	0.1715E-01	0.0	0.1115E 00	0.0	0.1021E-03
14	0.8938E 01	0.1700E-01	0.0	0.1106E 00	0.0	0.1021E-03
15	0.9014E 01	0.1686E-01	0.0	0.1097E 00	0.0	0.1021E-03
16	0.9090E 01	0.1672E-01	0.0	0.1087E 00	0.0	0.1021E-03
17	0.9166E 01	0.1658E-01	0.0	0.1078E 00	0.0	0.1021E-03
18	0.9242E 01	0.1644E-01	0.0	0.1069E 00	0.0	0.1021E-03
19	0.9319E 01	0.1631E-01	0.0	0.1061E 00	0.0	0.1021E-03
20	0.9394E 01	0.1618E-01	0.0	0.1052E 00	0.0	0.1021E-03
21	0.9470E 01	0.1605E-01	0.0	0.1044E 00	0.0	0.1021E-03
22	0.9546E 01	0.1592E-01	0.0	0.1035E 00	0.0	0.1021E-03
23	0.9622E 01	0.1579E-01	0.0	0.1027E 00	0.0	0.1021E-03
24	0.9698E 01	0.1567E-01	0.0	0.1019E 00	0.0	0.1021E-03
25	0.9774E 01	0.1555E-01	0.0	0.1011E 00	0.0	0.1021E-03
26	0.9850E 01	0.1543E-01	0.0	0.1003E 00	0.0	0.1021E-03
27	0.9926E 01	0.1531E-01	0.0	0.9958E-01	0.0	0.1021E-03
28	0.1000E 02	0.1519E-01	0.0	0.9882E-01	0.0	0.1021E-03
29	0.1008E 02	0.1508E-01	0.0	0.9807E-01	0.0	0.1021E-03
30	0.1015E 02	0.1497E-01	0.0	0.9734E-01	0.0	0.1021E-03
31	0.1023E 02	0.1486E-01	0.0	0.9662E-01	0.0	0.1021E-03

STATION	B STIFFNESS	D STIFFNESS	B PRIME	D PRIME
1	0.208163E 07	0.511471E 05	0.0	0.0
2	0.208163E 07	0.511471E 05	0.0	0.0
3	0.208163E 07	0.511471E 05	0.0	0.0
4	0.208163E 07	0.511471E 05	0.0	0.0
5	0.208163E 07	0.511471E 05	0.0	0.0
6	0.208163E 07	0.511471E 05	0.0	0.0
7	0.208163E 07	0.511471E 05	0.0	0.0
8	0.208163E 07	0.511471E 05	0.0	0.0
9	0.208163E 07	0.511471E 05	0.0	0.0
10	0.208163E 07	0.511471E 05	0.0	0.0
11	0.208163E 07	0.511471E 05	0.0	0.0
12	0.208163E 07	0.511471E 05	0.0	0.0
13	0.208163E 07	0.511471E 05	0.0	0.0
14	0.208163E 07	0.511471E 05	0.0	0.0
15	0.208163E 07	0.511471E 05	0.0	0.0
16	0.208163E 07	0.511471E 05	0.0	0.0
17	0.208163E 07	0.511471E 05	0.0	0.0
18	0.208163E 07	0.511471E 05	0.0	0.0
19	0.208163E 07	0.511471E 05	0.0	0.0
20	0.208163E 07	0.511471E 05	0.0	0.0
21	0.208163E 07	0.511471E 05	0.0	0.0
22	0.208163E 07	0.511471E 05	0.0	0.0
23	0.208163E 07	0.511471E 05	0.0	0.0
24	0.208163E 07	0.511471E 05	0.0	0.0
25	0.208163E 07	0.511471E 05	0.0	0.0
26	0.208163E 07	0.511471E 05	0.0	0.0
27	0.208163E 07	0.511471E 05	0.0	0.0
28	0.208163E 07	0.511471E 05	0.0	0.0
29	0.208163E 07	0.511471E 05	0.0	0.0
30	0.208163E 07	0.511471E 05	0.0	0.0
31	0.208163E 07	0.511471E 05	0.0	0.0

THE INITIAL CONDITIONS FOR N= 0 FOLLOW

STATION	U	V	W	M S
1	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0
6	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0
9	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0
13	0.0	0.0	0.0	0.0
14	0.0	0.0	0.0	0.0
15	0.0	0.0	0.0	0.0
16	0.0	0.0	0.0	0.0
17	0.0	0.0	0.0	0.0
18	0.0	0.0	0.0	0.0
19	0.0	0.0	0.0	0.0
20	0.0	0.0	0.0	0.0
21	0.0	0.0	0.0	0.0
22	0.0	0.0	0.0	0.0
23	0.0	0.0	0.0	0.0
24	0.0	0.0	0.0	0.0
25	0.0	0.0	0.0	0.0
26	0.0	0.0	0.0	0.0
27	0.0	0.0	0.0	0.0
28	0.0	0.0	0.0	0.0
29	0.0	0.0	0.0	0.0
30	0.0	0.0	0.0	0.0
31	0.0	0.0	0.0	0.0

STATION	U DOT	V DOT	W DOT	M S DOT
1	0.0	0.0	0.0	0.0
2	0.0	0.0	-0.141355E 04	0.0
3	0.0	0.0	-0.141355E 04	0.0
4	0.0	0.0	-0.141355E 04	0.0
5	0.0	0.0	-0.141355E 04	0.0
6	0.0	0.0	-0.141355E 04	0.0
7	0.0	0.0	-0.141355E 04	0.0
8	0.0	0.0	-0.141355E 04	0.0
9	0.0	0.0	-0.141355E 04	0.0
10	0.0	0.0	-0.141355E 04	0.0
11	0.0	0.0	-0.141355E 04	0.0
12	0.0	0.0	-0.141355E 04	0.0
13	0.0	0.0	-0.141355E 04	0.0
14	0.0	0.0	-0.141355E 04	0.0
15	0.0	0.0	-0.141355E 04	0.0
16	0.0	0.0	-0.141355E 04	0.0
17	0.0	0.0	-0.141355E 04	0.0
18	0.0	0.0	-0.141355E 04	0.0
19	0.0	0.0	-0.141355E 04	0.0
20	0.0	0.0	-0.141355E 04	0.0
21	0.0	0.0	-0.141355E 04	0.0
22	0.0	0.0	-0.141355E 04	0.0
23	0.0	0.0	-0.141355E 04	0.0
24	0.0	0.0	-0.141355E 04	0.0
25	0.0	0.0	-0.141355E 04	0.0
26	0.0	0.0	-0.141355E 04	0.0
27	0.0	0.0	-0.141355E 04	0.0
28	0.0	0.0	-0.141355E 04	0.0
29	0.0	0.0	-0.141355E 04	0.0
30	0.0	0.0	-0.141355E 04	0.0
31	0.0	0.0	0.0	0.0

THE INITIAL CONDITIONS FOR N= 1 FOLLOW

STATION	U	V	W	M S
1	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0
6	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0
9	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0
13	0.0	0.0	0.0	0.0
14	0.0	0.0	0.0	0.0
15	0.0	0.0	0.0	0.0
16	0.0	0.0	0.0	0.0
17	0.0	0.0	0.0	0.0
18	0.0	0.0	0.0	0.0
19	0.0	0.0	0.0	0.0
20	0.0	0.0	0.0	0.0
21	0.0	0.0	0.0	0.0
22	0.0	0.0	0.0	0.0
23	0.0	0.0	0.0	0.0
24	0.0	0.0	0.0	0.0
25	0.0	0.0	0.0	0.0
26	0.0	0.0	0.0	0.0
27	0.0	0.0	0.0	0.0
28	0.0	0.0	0.0	0.0
29	0.0	0.0	0.0	0.0
30	0.0	0.0	0.0	0.0
31	0.0	0.0	0.0	0.0

STATION	U DOT	V DOT	W DOT	M S DOT
1	0.0	0.0	0.0	0.0
2	0.0	0.0	-0.222040E 04	0.0
3	0.0	0.0	-0.222040E 04	0.0
4	0.0	0.0	-0.222040E 04	0.0
5	0.0	0.0	-0.222040E 04	0.0
6	0.0	0.0	-0.222040E 04	0.0
7	0.0	0.0	-0.222040E 04	0.0
8	0.0	0.0	-0.222040E 04	0.0
9	0.0	0.0	-0.222040E 04	0.0
10	0.0	0.0	-0.222040E 04	0.0
11	0.0	0.0	-0.222040E 04	0.0
12	0.0	0.0	-0.222040E 04	0.0
13	0.0	0.0	-0.222040E 04	0.0
14	0.0	0.0	-0.222040E 04	0.0
15	0.0	0.0	-0.222040E 04	0.0
16	0.0	0.0	-0.222040E 04	0.0
17	0.0	0.0	-0.222040E 04	0.0
18	0.0	0.0	-0.222040E 04	0.0
19	0.0	0.0	-0.222040E 04	0.0
20	0.0	0.0	-0.222040E 04	0.0
21	0.0	0.0	-0.222040E 04	0.0
22	0.0	0.0	-0.222040E 04	0.0
23	0.0	0.0	-0.222040E 04	0.0
24	0.0	0.0	-0.222040E 04	0.0
25	0.0	0.0	-0.222040E 04	0.0
26	0.0	0.0	-0.222040E 04	0.0
27	0.0	0.0	-0.222040E 04	0.0
28	0.0	0.0	-0.222040E 04	0.0
29	0.0	0.0	-0.222040E 04	0.0
30	0.0	0.0	-0.222040E 04	0.0
31	0.0	0.0	0.0	0.0

THE INITIAL CONDITIONS FOR N= 2 FOLLOW

STATION	U	V	W	M S
1	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0
6	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0
9	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0
13	0.0	0.0	0.0	0.0
14	0.0	0.0	0.0	0.0
15	0.0	0.0	0.0	0.0
16	0.0	0.0	0.0	0.0
17	0.0	0.0	0.0	0.0
18	0.0	0.0	0.0	0.0
19	0.0	0.0	0.0	0.0
20	0.0	0.0	0.0	0.0
21	0.0	0.0	0.0	0.0
22	0.0	0.0	0.0	0.0
23	0.0	0.0	0.0	0.0
24	0.0	0.0	0.0	0.0
25	0.0	0.0	0.0	0.0
26	0.0	0.0	0.0	0.0
27	0.0	0.0	0.0	0.0
28	0.0	0.0	0.0	0.0
29	0.0	0.0	0.0	0.0
30	0.0	0.0	0.0	0.0
31	0.0	0.0	0.0	0.0

STATION	U DOT	V DOT	W DOT	M S DOT
1	0.0	0.0	0.0	0.0
2	0.0	0.0	-0.942366E 03	0.0
3	0.0	0.0	-0.942366E 03	0.0
4	0.0	0.0	-0.942366E 03	0.0
5	0.0	0.0	-0.942366E 03	0.0
6	0.0	0.0	-0.942366E 03	0.0
7	0.0	0.0	-0.942366E 03	0.0
8	0.0	0.0	-0.942366E 03	0.0
9	0.0	0.0	-0.942366E 03	0.0
10	0.0	0.0	-0.942366E 03	0.0
11	0.0	0.0	-0.942366E 03	0.0
12	0.0	0.0	-0.942366E 03	0.0
13	0.0	0.0	-0.942366E 03	0.0
14	0.0	0.0	-0.942356E 03	0.0
15	0.0	0.0	-0.942366E 03	0.0
16	0.0	0.0	-0.942366E 03	0.0
17	0.0	0.0	-0.942366E 03	0.0
18	0.0	0.0	-0.942366E 03	0.0
19	0.0	0.0	-0.942366E 03	0.0
20	0.0	0.0	-0.942366E 03	0.0
21	0.0	0.0	-0.942366E 03	0.0
22	0.0	0.0	-0.942366E 03	0.0
23	0.0	0.0	-0.942366E 03	0.0
24	0.0	0.0	-0.942356E 03	0.0
25	0.0	0.0	-0.942366E 03	0.0
26	0.0	0.0	-0.942366E 03	0.0
27	0.0	0.0	-0.942366E 03	0.0
28	0.0	0.0	-0.942366E 03	0.0
29	0.0	0.0	-0.942366E 03	0.0
30	0.0	0.0	-0.942366E 03	0.0
31	0.0	0.0	0.0	0.0

THE INITIAL CONDITIONS FOR N= 4 FOLLOW

STATION	U	V	W	M S
1	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0
6	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0
9	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0
13	0.0	0.0	0.0	0.0
14	0.0	0.0	0.0	0.0
15	0.0	0.0	0.0	0.0
16	0.0	0.0	0.0	0.0
17	0.0	0.0	0.0	0.0
18	0.0	0.0	0.0	0.0
19	0.0	0.0	0.0	0.0
20	0.0	0.0	0.0	0.0
21	0.0	0.0	0.0	0.0
22	0.0	0.0	0.0	0.0
23	0.0	0.0	0.0	0.0
24	0.0	0.0	0.0	0.0
25	0.0	0.0	0.0	0.0
26	0.0	0.0	0.0	0.0
27	0.0	0.0	0.0	0.0
28	0.0	0.0	0.0	0.0
29	0.0	0.0	0.0	0.0
30	0.0	0.0	0.0	0.0
31	0.0	0.0	0.0	0.0

STATION	U DOT	V DOT	W DOT	M S DOT
1	0.0	0.0	0.0	0.0
2	0.0	0.0	0.188473E 03	0.0
3	0.0	0.0	0.188473E 03	0.0
4	0.0	0.0	0.188473E 03	0.0
5	0.0	0.0	0.188473E 03	0.0
6	0.0	0.0	0.188473E 03	0.0
7	0.0	0.0	0.188473E 03	0.0
8	0.0	0.0	0.188473E 03	0.0
9	0.0	0.0	0.188473E 03	0.0
10	0.0	0.0	0.188473E 03	0.0
11	0.0	0.0	0.188473E 03	0.0
12	0.0	0.0	0.188473E 03	0.0
13	0.0	0.0	0.188473E 03	0.0
14	0.0	0.0	0.188473E 03	0.0
15	0.0	0.0	0.188473E 03	0.0
16	0.0	0.0	0.188473E 03	0.0
17	0.0	0.0	0.188473E 03	0.0
18	0.0	0.0	0.188473E 03	0.0
19	0.0	0.0	0.188473E 03	0.0
20	0.0	0.0	0.188473E 03	0.0
21	0.0	0.0	0.188473E 03	0.0
22	0.0	0.0	0.188473E 03	0.0
23	0.0	0.0	0.188473E 03	0.0
24	0.0	0.0	0.188473E 03	0.0
25	0.0	0.0	0.188473E 03	0.0
26	0.0	0.0	0.188473E 03	0.0
27	0.0	0.0	0.188473E 03	0.0
28	0.0	0.0	0.188473E 03	0.0
29	0.0	0.0	0.188473E 03	0.0
30	0.0	0.0	0.188473E 03	0.0
31	0.0	0.0	0.0	0.0

THE TIME STEP NUMBER IS 250   THE TIME IS 4.5? OF 0.500E-03 SECONDS   THE SOLUTION CONVERGED IN 2 ITERATIONS

THE SUMMED FORCES, MOMENTS, DISPLACEMENTS AND ROTATIONS FOLLOW FOR THETA = 0.0

STATION	N S	N THETA	N STHETA	Q S	M S	M THETA	M STHETA
1	-0.3459E 04	-0.9571E 03	0.0	-0.2372E 04	0.2386E 03	0.5924E 02	0.0
14	0.7629E 03	0.2451E 04	0.0	0.2591E 04	0.1838E 04	0.4448E 03	0.0
27	0.3078E 04	-0.3224E 04	0.0	-0.2860E 04	-0.6155E 03	-0.1715E 03	0.0
31	0.2631E 04	0.7875E 03	0.0	0.1322E 04	-0.9756E 02	-0.2790E 02	0.0

STATION	U	V	W	PHI S	PHI THETA	PHI
1	0.0	0.0	0.2249E-09	0.1778E-08	0.0	0.0
14	-0.6017E-02	0.0	0.1312E 00	-0.3545E-01	0.0	0.0
27	-0.4079E-02	0.0	0.3927E-01	0.4481E-01	0.0	0.0
31	-0.1578E-09	0.0	-0.6352E-10	-0.5080E-09	0.0	0.0

THE SUMMED FORCES, MOMENTS, DISPLACEMENTS AND ROTATIONS FOLLOW FOR THETA = 0.3141591 01

STATION	N S	N THETA	N STHETA	Q S	M S	M THETA	M STHETA
1	0.4580E 04	0.1275E 04	-0.3960E-02	-0.1934E 04	0.1454E 04	0.4158E 03	-0.1914E-04
14	-0.6554E 04	-0.3291E 05	-0.1549E-02	-0.2561E 03	-0.8858E 03	-0.3320E 03	0.1056E-03
27	-0.3588E 04	-0.6111E 04	0.4616E-02	0.3640E 03	0.5826E 03	0.1065E 03	-0.7260E-04
31	-0.1638E 04	-0.5050E 03	0.3203E-02	0.5499E 03	0.1458E 04	0.4283E 03	0.1140E-04

STATION	U	V	W	PHI S	PHI THETA	PHI
1	0.0	0.0	-0.4935E-09	0.3810E-08	0.1078E-15	-0.2664E-08
14	0.5233E-02	-0.3805E-07	-0.2074E 00	0.2529E-01	0.1766E-07	-0.2672E-08
27	0.2689E-02	-0.1181E-07	-0.4600E-01	-0.3238E-01	0.4165E-08	0.4195E-08
31	0.3275E-10	-0.4139E-15	-0.3176E-10	0.2540E-09	-0.4097E-16	0.2155E-08

APPENDIX C

INPUT DATA GUIDE FOR SATANS-IIA

INPUT DATA GUIDE FOR SATANS-I, SATANS-II, AND SATANS-IIA

CARD	COLUMNS	FORMAT	ITEM	EXAMPLE	MEANING
1	1-72	18A4	TITLE	-	ENTER ANY 72 CHARACTERS
2	1-5	I5	NO	1	THE PROBLEM NUMBER, 0<NO<10000.
2	6-10	L5	$DYNAMC	F	FOR A STATIC ANALYSIS, SET $DYNAMC = F.
				T	FOR A DYNAMIC ANALYSIS, SET $DYNAMC = T.
2	11-15	I5	IMODE	0	FOR NO MODAL OUTPUT DATA
				1	FOR MODAL OUTPUT DATA.
2	16-20	I5	NDIMEN	0	DIMENSIONAL OUTPUT DATA.
				1	NONDIMENSIONAL OUTPUT.
2	21-25	I5	NTHMAX	8	SUMMED SOLUTION WILL BE PRINTED AT NTHMAX MERIDIANS, 0<=NTHMAX<=36.
2	26-30	I5	IFREQ	2	SOLUTION WILL BE PRINTED AT THE FIRST STATION, EVERY SUBSEQUENT IFREQ STATION AND THE LAST STATION, 0<IFREQ<=KMAX.
2	31-35	I5	IPRINT	3	EVERY IPRINT CONVERGED SOLUTION WILL BE PRINTED.
2	36-40	I5	IBCINL	-1	IF THE SHELL HAS A POLE AT THE FIRST STATION.
				0	IF THE SHELL HAS NO POLE AT THE FIRST STATION.
2	41-45	I5	IBCFNL	-1	IF THE SHELL HAS A POLE AT THE LAST STATION.
				0	IF THE SHELL HAS NO POLE AT THE LAST STATION.

CARD	COLUMNS	FORMAT	ITEM	EXAMPLE	MEANING
2	46-50	I5	KMAX	35	NUMBER OF MERIDIONAL STATIONS. NOTE: KMAX<201 FOR SATANS-I WITHOUT PLOTS AND KMAX<101 FOR SATANS-I WITH PLOTS OR FOR SATANS-II. SATANS-IIA IS UNLIMITED.
2	51-55	I5	MNMAX	7	NUMBER OF SERIES COEFFICIENTS USED TO DESCRIBE THE INITIAL CONDITIONS, PRESSURE AND THERMAL LOADS (AND INITIAL IMPERFECTIONS IF USING SATANS-II OR IIA). MNMAX<=MAXM.
2	56-60	I5	MAXM	7	MAX NUMBER OF HARMONICS IN THE SOLUTION, LIMITED TO 99.
2	61-65	I5	LSMAX	1	FOR A LINEAR ANALYSIS.
				99	USE MANY LOAD STEPS FOR A NONLINEAR STATIC ANALYSIS.
				3000	FOR A DYNAMIC ANALYSIS, LSMAX IS THE NUMBER OF TIME INCREMENTS, WHERE LSMAX= T MAX/ΔT.
2	66-70	I5	LCHMAX	2	THE NUMBER OF LOAD STEP SIZE REDUCTIONS IN A STATIC ANALYSIS, RECOMMENDED RANGE = 2-4.
				0	FOR A DYNAMIC ANALYSIS.
2	71-75	I5	ITRMAX	1	FOR A LINEAR ANALYSIS.
				30	THE NUMBER OF ITERATIONS AT A LOAD OR TIME STEP. FOR A NONLINEAR ANALYSIS, SUGGESTED RANGE = 10-30, UP TO 50 FOR SPECIAL CASES.
2	76-80	I5	IC	0	INITIAL CONDITIONS. SET TO 0 FOR A STATIC ANALYSIS, OR FOR A DYNAMIC ANALYSIS WHERE THE SHELL IS AT REST AT T= 0.
				1	FOR A DYNAMIC ANALYSIS WITH INITIAL CONDITIONS.

CARD	COLUMNS	FORMAT	ITEM	EXAMPLE	MEANING
3	1-12	E12.3	NU	0.3	POISSON'S RATIO, ν.
3	12-24	E12.3	SIGO	1000.0	REFERENCE STRESS LEVEL, σ_0.
				1.0	IF THE INPUT DATA IS DIMENSIONAL.
3	24-36	E12.3	ELAST	.3E8	REFERENCE MODULUS OF ELASTICITY, E_0.
				1.0	IF THE INPUT DATA IS DIMENSIONAL.
3	37-48	E12.3	TKN	.4E-2	REFERENCE THICKNESS, h_0.
				1.0	IF THE INPUT DATA IS DIMENSIONAL.
3	49-60	E12.3	CHAR	8.16	CHARACTERISTIC SHELL DIMENSION, a.
				1.0	IF THE INPUT DATA IS DIMENSIONAL.
3	61-72	E12.3	TEEO	0.0	IF A STATIC ANALYSIS.
				.996E-5	REFERENCE TIME, T_0.
4	1-12	E12.3	DELOAD	0.2	FOR A STATIC ANALYSIS, DELOAD IS THE LOAD INCREMENT. IT REMAINS UNCHANGED UNTIL THE SOLUTION FAILS TO CONVERGE IN ITERMAX ITERATIONS, WHEN IT IS REDUCED BY A FACTOR OF FIVE. A MAXIMUM OF LCHMAX SUCH REDUCTIONS WILL OCCUR.
				.1823E-6	FOR A DYNAMIC ANALYSIS, DELOAD IS THE NONDIMENSIONAL TIME INCREMENT.
4	13-24	E12.3	EPS	0.01	THE CONVERGENCE CRITERION RECOMMENDED RANGE OF 0.01<EPS<0.001.

CARD 4A IS ONLY INCLUDED FOR A SATANS-II OR SATANS-IIA RUN.

4A	1-5	I5	JUMP	1	FOR AN ANALYSIS USING SINGLE SERIES EXPANSIONS.
				2	FOR AN ANALYSIS USING DOUBLE SERIES EXPANSIONS.
4A	5-10	I5	MPERFS	0	AN ANALYSIS WITHOUT IMPERFECTIONS.
				1	AN ANALYSIS WITH IMPERFECTIONS. NOTE: IF JUMP=2, MPERFS MAY BE 0 OR 1. IF JUMP =1, MPERFS MUST BE 0. IF MPERFS=1, JUMP MUST BE 2.

CARC	CCLUMN	FCRMAT	ITEM	EXAMPLE	MEANING

INCLLDE AS MANY CARDS 5 AS NECESSARY TC SPECIFY NTHMAX MERICIANS. IF NTHMAX EQUALS 0, OMIT CARC 5.

5	1-72	6E12.3		10.0	A LIST OF CIRCLMFERENTIAL COCRCINATES θ, IN CEGREES ANC TENTHS, WHERE THE SCLUTICN PRINTCUT IS CE-SIREC. THE LIST MUST HAVE NTHMAX ENTRIES.

IF IBCINL= -1, CMIT CARDS 6 THROUGH 14. IF IBCFNL= -1, CMIT CARCS 15 THROUGH 23. CARDS 6 THROUGH 23 DESCRIBE THE BCUNCARY CCNDITICNS AT THE FIRST, ANC THEN AT THE LAST STA-TICN. THE BOUNDARY CONDITIONS EXIST CN THE TOTAL VARIABLES, NCT CN THE INCIVIDUAL HARMONICS. LOACINGS APPLIEC THROUGH SPECIFICATION CF BOUNDARY CCNDITICNS ARE TAKEN IN THE ZERC-ETH HARMCNIC (N=0) ONLY, AS THE COLUMN MATRIX $\{\ell\}$ IS SET TC ZERC FOR HARMCNICS GREATER THAN ZERO. THE BOUNCARY CON-DITICNS ARE DIMENSIONAL. THE FORMAT CF CARDS 6 THRCUGH 23 IS 4E16.8.

CARC 6,15 CARD 7,16 CARD 8,17 CARD 9,18

$$\begin{bmatrix} \Omega(1,1) & \Omega(1,2) & \Omega(1,3) & \Omega(1,4) \\ \Omega(2,1) & \Omega(2,2) & \Omega(2,3) & \Omega(2,4) \\ \Omega(3,1) & \Omega(3,2) & \Omega(3,3) & \Omega(3,4) \\ \Omega(4,1) & \Omega(4,2) & \Omega(4,3) & \Omega(4,4) \end{bmatrix} \begin{bmatrix} N_s \\ N_{s\theta} \\ Q_s \\ \phi_s \end{bmatrix} +$$

CARC 10,19 CARD 11,20 CARD 12,21 CARC 13,22 CARC 14,23

$$\begin{bmatrix} \Lambda(1,1) & \Lambda(1,2) & \Lambda(1,3) & \Lambda(1,4) \\ \Lambda(2,1) & \Lambda(2,2) & \Lambda(2,3) & \Lambda(2,4) \\ \Lambda(3,1) & \Lambda(3,2) & \Lambda(3,3) & \Lambda(3,4) \\ \Lambda(4,1) & \Lambda(4,2) & \Lambda(4,3) & \Lambda(4,4) \end{bmatrix} \begin{bmatrix} U \\ V \\ W \\ M_s \end{bmatrix} = \begin{bmatrix} \ell(1) \\ \ell(2) \\ \ell(3) \\ \ell(4) \end{bmatrix}$$

CARC 24 IS:

1. INCLUDED FOR A SATANS-I STATIC ANALYSIS.
2. INCLUDED BUT BLANK FCR A SATANS-I DYNAMIC ANALYSIS.
3. CMITTEC FCR A SATANS-II ANALYSIS.
4. INCLUDEC BLANK FOR DYNAMIC USED FCR STATIC SATANS-IIA ANALYSES.

CARC	CCLUMN	FCRMAT	ITEM	EXAMPLE	MEANING
24	1-2	L2	$PLOTS	F	INDICATES PLCTS ARE NCT DESIRED.
				T	INCICATES PLCTS ARE DESIRED.
24	3-4	L2	$MCDAL	F	INDICATES PLCTS ARE FCR SUMMED SOLUTICNS ONLY.
				T	INCICATES PLCTS ARE FCR MODAL SOLUTICNS CNLY.

FOR THE REMAINDER OF CARD 24 ENTRIES, 0 INDICATES THAT NO PLOTS ARE DESIRED FOR THE PARTICULAR ITEM, AND 1 INDICATES THAT THEY ARE DESIRED. ALL GRAPHS ARE PLOTTED AS THE INDICATED ITEM VERSUS THE STATION NUMBER. IF A COMPLETE PLOT IS DESIRED, INSUTE IFREQ = 1.

CARD	COLUMN	FORMAT	ITEM	EXAMPLE	MEANING
24	5- 6	I2	IRADII	1	PLOT THE RADII AS COMPUTED BY SUBROUTINE GEOM.
24	7- 8	I2	IGAMMA	1	PLOT ρ'/ρ AS COMPUTED BY SUBROUTINE GEOM.
24	9-10	I2	IOMEGS	1	PLOT ω_s AS COMPUTED BY SUBROUTINE GEOM.
24	11-12	I2	IOMEGT	1	PLOT ω_θ AS COMPUTED BY SUBROUTINE GEOM.
24	13-14	I2	IDECMS	1	PLOT ω_s' AS COMPUTED BY SUBROUTINE GEOM.
24	15-16	I2	IBSTIF	1	PLOT THE STIFFNESS D AS COMPUTED BY SUBROUTINE BDB.
24	17-18	I2	IDSTIF	1	PLOT THE STIFFNESS D AS COMPUTED BY THE SUBROUTINE BDB.
24	19-20	I2	IBBSTF	1	PLOT THE STIFFNESS db/ds AS COMPUTED BY SUBROUTINE BDB.
24	21-22	I2	IDDSTF	1	PLOT THE STIFFNESS dd/ds AS COMPUTED BY SUBROUTINE BDB.
24	23-24	I2	IPR	1	PLOT THE NORMAL COMPONENT OF THE PRESSURE LOAD.
24	25-26	I2	IPS	1	PLOT THE MERIDIONAL COMPONENT OF THE PRESSURE LOAD.
24	27-28	I2	IPT	1	PLOT THE CIRCUMFERENTIAL COMPONENT OF THE PRESSURE LOAD.
24	29-30	I2	ITT	1	PLOT THE THERMAL LOAD.
24	31-32	I2	IMT	1	PLOT THE THERMAL MOMENT.
24	33-34	I2	IDTT	1	PLOT d/ds OF THE THERMAL LOAD.
24	35-36	I2	IDMT	1	PLOT d/ds OF THE THERMAL MOMENT.
24	37-38	I2	INS	1	PLOT THE MERIDIONAL MEMBRANE FORCE DISTRIBUTION.

CARD	COLUMN	FORMAT	ITEM	EXAMPLE	MEANING
24	39-40	I2	INTH	1	PLOT THE CIRCUMFERENTIAL MEMBRANE FORCE DISTRIBU-TION.
24	41-42	I2	INSTH	1	PLOT THE MERIDIO-CIRCUMFERENTIAL MEMBRANE FORCE DISTRIBUTION.
24	43-44	I2	IQS	1	PLOT THE TRANSVERSE FORCE DISTRIBUTION.
24	45-46	I2	IMS	1	PLOT THE MERIDIONAL MOM-ENT DISTRIBUTION.
24	47-48	I2	IMTH	1	PLOT THE CIRCUMFERENTIAL MOMENT DISTRIBUTION.
24	49-50	I2	IMSTH	1	PLOT THE MERIDIO-CIRCUMFERENTIAL MOMENT DISTRIBUTION.
24	51-52	I2	IU	1	PLOT THE MERIDIONAL DIS-PLACEMENT DISTRIBUTION.
24	53-54	I2	IV	1	PLOT THE CIRCUMFERENTIAL DISPLACEMENT DISTRI-BUTION.
24	55-56	I2	IW	1	PLOT THE NORMAL DISPLACE-MENT DISTRIBUTION.
24	57-58	I2	IPHIS	1	PLOT THE MERIDIONAL ROTA-TION DISTRIBUTION.
24	59-60	I2	IPHIT	1	PLOT THE CIRCUMFERENTIAL ROTATION DISTRIBUTION.
24	61-62	I2	IPHI	1	PLOT THE MERIDIO-CIRCUMFERENTIAL ROTATION DISTRIBUTION.

INSERT IMPERFECTION DATA HERE FOR A SATANS-II OR SATANS-IIA ANALYSIS WITH IMPERFECTIONS. INSURE FORMAT OF THE IMPERFEC-TION DATA IS COMPATIBLE WITH THAT SPECIFIED IN THE USER-WRITTEN SUBROUTINE IMPERF.

CARD	COLUMN	FORMAT	ITEM	EXAMPLE	MEANING
25	1-2	I2	IRNAGN	0	INDICATES THIS IS THE ONLY RUN.
				1	INDICATES ANOTHER RUN IS TO BE MADE. ADD ANOTHER COMPLETE SET OF DATA CARDS AFTER THIS CARD IS IRNAGN= 1.

APPENDIX D

LISTING OF NEW POLE ROUTINE FOR SATANS-IIA

THE FOLLOWING CARDS ARE TO BE PLACED INTO THE FORCE SUBROUTINE

```
      COMMON /IBL5/IBCINL,IBCFNL
```

```
C    IN FORCE
   10 IF(K.NE.2.OR.(K.EQ.2.AND.IBCINL.GE.0)) GO TO 901
      DO 902 II=1,4
      SUMX=0.
      DO 903 L=1,4
  903 SUMX=SUMX+DL(II,L,M)*GEE(L)
  902 X(II,IK1)=SUMX
  901 CONTINUE
      DO 11 I=1,4
```

THE FOLLOWING CARDS ARE TO BE PLACED INTO THE PMATRX SUBROUTINE

```
C    IN PMATRX
      CALL EFG(2,MN)
      CALL ABC
      CALL MATINV(A,4,G1,0,DETERM,IPIVCT,INDEX,4,ISCALE)
      DO 901 II=1,4
      DO 901 JJ=1,4
      DL(II,JJ,MN)=0.
      DG(II,JJ,MN)=0.
  901 DF(II,JJ,MN)=0.
      IF(NN.GT.1) GO TO 12
      IF(NN.GT.0) GO TO 13
```

```
      MC=MN
      DL(1,1,MN)=1.
      DL(2,2,MN)=1.
      DL(3,3,MN)=-3.
      DL(4,4,MN)=-3.
      DG(3,3,MN)=4.
      DG(4,4,MN)=4.
      DF(3,3,MN)=-1.
      DF(4,4,MN)=-1.
      GO TO 902
   13 M1=MN
      DL(1,1,MN)=-3.
      DL(2,1,MN)=1.
      DL(2,2,MN)=1.
      IF(N(M1).LT.0) DL(2,2,MN)=-1
      DL(3,3,MN)=1.
      DL(4,4,MN)=1.
      DG(1,1,MN)=4.
      DF(1,1,MN)=-1.
      GO TO 902
   12 M2=MN
      DL(1,1,MN)=1.
      DL(2,2,MN)=1.
      DL(3,3,MN)=1.
      DL(4,4,MN)=-3.
      DG(4,4,MN)=4.
      DF(4,4,MN)=-1.
  902 CONTINUE
      DO 903 II=1,4
      DO 903 JJ=1,4
      TTP=0.
      DO 904 L=1,4
  904 TTP=TTP+DF(II,L,MN)*A(L,JJ)
  903 CLO(II,JJ)=TTP
      DO 905 II=1,4
      DO 905 JJ=1,4
      TTP=0.
      TTQ=0.
      DO 906 L=1,4
      TTP=TTP+CLO(II,L)*C(L,JJ)
  906 TTQ=TTQ+CLO(II,L)*BEE(L,JJ)
      CL1(II,JJ)=DL(II,JJ,MN)-TTP
  905 CL2(II,JJ)=DG(II,JJ,MN)-TTQ
      CALL MATINV(CL1,4,G1,0,DETERM,IPIVOT,INDEX,4,ISCALE)
      DO 907 II=1,4
      DO 907 JJ=1,4
      TTP=0.
      TTQ=0.
```

```
      DO 508 L=1,4
      TTP=TTP+CL1(II,L)*CL0(L,JJ)
  508 TTQ=TTQ+CL1(II,L)*CL2(L,JJ)
      CL(II,JJ,MN)=-TTP
  507 P(II,JJ,IJ)=TTQ
      GO TO 11
   50 M3=MN
```

APPENDIX E

LISTING OF CARDS FOR $\bar{V}$ AND $\bar{V}_{MAX}$

THE FOLLOWING CARDS ARE TO BE PLACED INTO THE DYNAMIC SUBROUTINE IF NEEDED

```
C     STATEMENTS FOR MAIN TO CALCULATE VBAR
  185 DENOM=.125*GMXI(KMAX)*R(KMAX)**4
      DO 186 M=1,MAXM
      DNUM=0.
      MM=(M-1)*KMAX2
      DO 184 K=2,KL
      KT=K+1+MM
  184 DNUM=DNUM+Z(3,KT)*R(K)
      DNUM=DNUM*DEL*SOE
  186 VBAR(M)=DNUM/DENOM
      ITTEST=ITTEST+1
      IF(ITTEST.NE.10) GO TO 963
      ITTEST=0
      WRITE(6,183)(LSTEP,(VBAR(M),M=1,MAXM))
  183 FORMAT(/5X,'VBAR AT TIME STEP ',I4,' FOR EACH MODE IS'/5E16.4)
  963 CONTINUE
      DO 187 M=1,MAXM
      IF(LSTEP.EQ.1) AVB(M)=0.
  187 IF(ABS(VBAR(M)).GT.AVB(M)) AVB(M)=ABS(VBAR(M))
```

LIST OF FIGURES AND TABLES

A. FIGURES

B. TABLES

LIST OF REFERENCES

1. Ball, R. E. and Burt, J. A., " Dynamic buckling of shallow spherical shells", Journal of Applied Mechanics, v. Paper No. 73-APM-A, p. 411 to 416, June 1973.

2. Stilwell, W. C., and Ball, R. E., "A Digital Computer Study of the Buckling of Shallow Spherical Caps and Truncated Hemispheres,"NASA CR-1998, June 1972.

3. Ball R. E., "A Program for the Nonlinear Static and Dynamic Analysis of Arbitrarily Loaded Shells of Revolution," Cumputers and Structures, Vol. 2,1972, p. 141 to 162; also "A Computer Program for the Geometrically Nonlinear Static and Dynamic Analysis of Arbitrarily Loaded Shells of Revolution, Theory, and Users Manual," NASA CR-1987, April 1972.

4. Huang, N. C., " Unsymmetrical Buckling of Thin Shallow Spherical Shells", Journal of Applied Mechanics, v. 31, p. 447 to 457, September 1964.

5. Huang, N. C., " Axisymmetric Dynamic Snap-Through of Elastic Clamped Shallow Spherical Caps Due to Centrally Distributed Pressures", AIAA Journal, v. 7, No. 2, p. 215 to 220, Feb. 1969.

6. Stephens, W. B., and Fulton, R. E., " Axisymmetric Static and Dynamic Buckling of Spherical Caps Due to Centrally Distributed Pressures", AIAA Journal, v. 7, No. 11, p. 2120 to 2126, November 1969.

7. Lock, M. H., Okubo, S. and Whittier, J. S.,

"Experiments on the Snapping of a Shallow Dome under a Step Pressure Load", AIAA Journal, v. 6, No. 1, p. 1320 to 1326, July 1968.

8. Stricklin, J. A., et al., " Nonlinear Dynamic Analysis of Shells of Revolution by Matrix Displacement Method", AIAA Journal, v. 9, No. 4, p. 629 to 636, April 1971.

9. Akkas, N., " Bifercation and Snap-Through Phenomena in Asymmetric Dynamic Analysis of Shallow Spherical Shells", Computers and Structures, v. 6, p. 241 to 251, March 1976.

10. Ryan, B. A., "A Digital Computer Study of the Buckling of Actual Imperfect Cylinders - A Modification of the Computer Program SATANS - Theory and User's Manual for SATANS- I and SATANS-II," A. E. Thesis, Naval Postgraduate School, June 1970.

11. Ball, R. E. and Ryan, B. A., " Computer Analysis of Buckling of Imperfect Shells", Journal of the Structural Division, v. 99, No. ST10, p. 2097 to 2108, October 1973.

12. Ball, R. E. et al., " A Comparison of Computer Results for the Dynamic Response of the LMSC Truncated Cone", Computers and Structures, v. 4, p. 485 to 498, July 1973.

13. Aerospace Structures Information and Analysis Center, AFFDL/FBR, Wright-Patterson AFB, Ohio 45433